

# PhotosynQ – cloud platform powered by IoT devices

Andriy Herts, Ivan Tsidylo\*, Nataliia Herts, Ljubov Barna, and Stanislav-Ivan Mazur

Ternopil Volodymyr Hnatiuk National Pedagogical University, 2 Maxyma Kryvonosa Str., Ternopil, 46027, Ukraine

**Abstract.** This article describes how to connect to PhotosynQ and some of the features of a cloud service. The focus is on organizing communication between the web service and the Internet of Things. An example and procedure for integrating a number of digital encoders that are not included in the list supported by the default platform through modification of the open-source base firmware is given. It has been suggested that in the future, a number of tools can be designed in the future, including for environmental monitoring, using basic digital sensors included in the so-called Arduino UNO Kit kits and more. It is emphasized that the PhotosynQ web platform, which is related to the SaaS cloud computing model, is an environment for collecting, analyzing, deriving and discussing results where research protocols and calculation procedures are open access. Its primary focus is only on working with a specific list of dashboards, offset by the openness of the software itself, which provides communication with the platform.

## 1 Introduction

Now there is a growing need for cloud services, computing. Cloud technology is becoming an integral part of many companies and according to IDC forecasts [1], up to 67% of IT infrastructure and software will be implemented with its use.

Over the last 5 years, they have changed not only the IT industry. Ninety percent of US educational institutions have switched to cloud technology [2]. Scientific institutions of the world are no exception.

A variety of web-based software tools used in large-scale science applications help with computation, data integration, analysis, and process automation. Using information obtained through computational procedures, they help scientists to solve applied problems and facilitate the dissemination of data. Here, at least, one should mention the complex research in the field of bioinformatics, the study of brain biology, using network and / or web-based tools for the work process [3].

Cloud technologies and services based on them, in particular PaaS, SaaS, IaaS [4], are gradually becoming platforms through which one can explore the environment. Based on these solutions, researchers can build the tools they need to build databases, making all the settings remotely via the Internet.

Of course, today we do not have to talk about a universal cloud service that would satisfy the needs of the scientific community as a whole, and the need to simultaneously use all three of the key aforementioned services that cloud services provide. The diversity of the data set, their heterogeneity, the differences in the amount of data sets, and therefore the need for different computing capacities, the lack of a unified interface for communication and data exchange all require individual cloud service architecture for the specific tasks of users

and researchers. However, the main advantages of using cloud platforms are as follows: speed of creation of new applications, flexibility and scalability of the system.

With the development of RFID (Radio Frequency Identification), WSN (Wireless Sensor Network), NFC (Near Field Communication) and M2M (Machine-to-Machine), which, when integrated with the Internet, allow for easy communication of various technical devices (“things”), another type of platform, namely IoT cloud platforms, is gaining in popularity.

IoT allows you to create a combination of smart devices interconnected by communication networks, and people. Together, they can create a variety of systems, for example, to work in environments that are inaccessible to humans, etc.

With the development of IoT, more objects (“things”) will connect to the global network. In this way, new opportunities will emerge in various spheres of human activity, and new perspectives will be opened to improve the quality of life of the population, in particular by improving the food security of mankind. There is every prospect that in the future, “things” will become active participants in business, information and social processes, where they will be able to interact and communicate with each other, sharing information about the environment, reacting and influencing the processes occurring in the surrounding world.

As platforms grow, so does the need to explore their capabilities and the tools they provide. One of the cloud platforms that is designed to produce exceptionally high-quality field-based scientific results is PhotosynQ [5].

PhotosynQ is first and foremost a network of open Internet data on the status of plant organisms and their phenotyping [5]. Basically, the primary data is obtained through the operation of a tool, a device [6] that captures information from plant objects. The latter is based on the

\* Corresponding author: [tsidylo@tntpu.edu.ua](mailto:tsidylo@tntpu.edu.ua)

chlorophyll fluorescence parameters and characterizes the course of the primary photosynthesis processes. The final data are obtained by processing the primary data and visualizing them with on-line platforms.

This creates a communication platform that connects a large number of researchers, groups of researchers, breeders, farmers, teachers and civilian scientists, creating new opportunities in the direction of research management, raising the level of knowledge about the phenomena and objects under study. Existing PhotosynQ projects are designed to remove barriers between laboratory and field research through the development of tools to study, including the state of plants in dynamic natural environments, etc. [7].

The platform itself is easy to use. According to the developers, it allows not only different users to exchange, compare and analyze the results of the research, but also in its specific setting, to take into account the specifics of the research, to easily adapt the researcher to their own needs.

The stated principles of operation of the said platforms [8] comply with the basic principles of the Internet of Things [9], the key of which is the ability of each object to send and receive data via a personal network or the Internet. This makes the PhotosynQ platform cloud-oriented and capable of communicating with things.

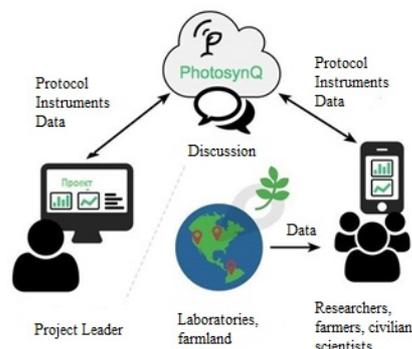
## 2 Analysis of recent research and publications

Plant phenotyping [8] is first and foremost the main goal of the cloud-oriented PhotosynQ platform. Over the past few years, researchers from more than 25 countries have been using it actively, making hundreds of thousands of measurements. A number of different projects on plant breeding, agronomy and physiology have been obtained and uploaded [8]. Frequent sources [10] inform about the use of PhotosynQ for educational purposes. For instance, Delaware Valley University (USA) student research course, which is part of the University's Experience360 program and is aimed at providing students with practical research experience in the field of biology, is based on the application of this online platform [11]. In total, as of October 2018, the total number of measurements received and processed by PhotosynQ reached one million.

However, one should pay attention to its capabilities in the direction of exchange, data analysis, the ability to customize to meet the needs of both researchers and users in general. In the long run, this can bring it out of the scope of the use only by plant object researchers and provide a tool for collaborative research, such as the natural environment as a whole.

Figure 1 shows the architecture of the PhotosynQ cloud platform [8], which was developed at Michigan State University (Michigan State University, MSU, USA) [5]. The primary and leading role in this scheme is played by devices that can collect all kinds of information, data and disseminate them over communication networks. Therefore, the "thing" in our

case is represented by the "device" MultispeQ [8]. This is not in contradiction with Recommendation ITU-TY.2060 [12], concerning the definition of the Internet of Things, since the said device is capable of communication.



**Fig. 1.** The principle of organization of the PhotosynQ network [8]

With regard to communication itself, it is known that many network technologies are provided for IoT such as global networks, local area networks, self-organizing wireless networks (ad-hoc) and mesh networks [13]. These communication networks transfer the data collected by the devices to the respective software applications, and transmit commands from the software applications to the devices. Above, we [14] characterized ThingSpeak cloud service as a means of training biology students to monitor and estimate the air pollution of the surface layer of the atmosphere. The prototype that provides monitoring of the human respiratory zone in two districts of the city of Ternopil on the content of suspended particulate matter PM10 and PM2.5 was developed. The main components of the instrument were implemented via microcontroller development system – Teensy 3.2, sensor module (temperature, humidity, pressure) BME280, SenseAir S8 carbon dioxide sensor module, PMS3003 air pollution sensor, Wi-Fi module ESP-01 and online ThingSpeak platform for storing and processing data.

In the case of PhotosynQ, communication between the device and the online platform is accomplished through the mobile PhotosynQ App or its Desktop version [8] using Bluetooth technology, in particular. The power of a mobile device with the appropriate application installed is not only involved in the management and collection of data, but is also used for initial calculations. In the end, the application sends the data to the cloud.

Analyzing the literature, documentation and information support of the site [8], we note that the developers were guided by Open Hardware principles when designing a tool integrated with the PhotosynQ cloud platform [15]. Open access contains documentation on its use and describes the main specifications [7]. The procedure for creating protocols for the study organization is described in detail, the procedure for creating protocols (instructions) for device control is described as well [8]. The latter involve access to a single sensor, both the structural component of the

device and all sensors in general. Given the number of encoders (sensors) that is supported by the firmware of the device, which was created by current owner and project participant Our Sci Greg Austic [16] and Jon Zeef, we are able to analyze a fairly wide range of physical parameters of the environment or objects, whenever necessary.

According to the developers [8, 17], the device is designed so that if necessary, it can integrate additional necessary extensions, sensors, etc. The core of the “thing” is the Teensy microcontroller [18] that controls all components, sensors. Given that open-source firmware [9] is a principle (specify the type of license), it creates the ability to modify the tool, “thing” for one’s own needs.

Therefore, a sufficient condition for communication with the PhotosynQ on-line platform is a Teensy microcontroller with appropriate firmware [17]. Its analysis seems to suggest that, subject to certain, sometimes not very significant changes, support for a much wider range of intelligent sensors (transmitters) can be obtained. Thus, to get a simple tool that will significantly expand the research or teaching tool of a researcher or a teacher accordingly.

### 3 The results of the research

The hardware to communicate with the PhotosynQ platform.

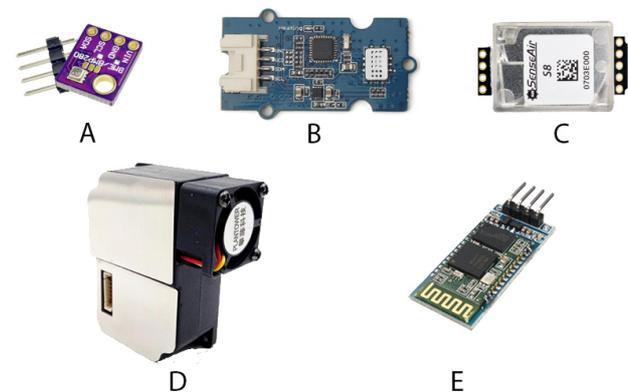
Our measurement system consists of intelligent, digital sensors (DS), measuring converters (see Fig. 2), which are connected to the signal processing system. The latter is represented by the Teensy 3.2 hardware and computing platform [18]. The hardware, according to its technical characteristics, is able to collect temperature data (−40 to +85 °C, ±0.5 °C), relative humidity (0-100%, ± 3%), atmospheric pressure (300–1100 hPa, ±1.0hPa), carbon dioxide (CO<sub>2</sub>) level (0.04% to 2%), PM concentration (PM 2.5, PM 10, µg/m<sup>3</sup>), (measurements range from 0.3 to 1.0; from 1.0 to 2.5; from 2.5 to 10 microns) in the air, the level (ppm) of carbon monoxide (CO) (1–1000 ppm), nitrogen dioxide (NO<sub>2</sub>) (0.05–10 ppm), ethanol (C<sub>2</sub>H<sub>5</sub>OH) (10–500 ppm), hydrogen (H<sub>2</sub>) (1–1000 ppm), ammonia (NH<sub>3</sub>) (1–500 ppm) methane (CH<sub>4</sub>) (> 1000 pmm), propane C<sub>3</sub>H<sub>8</sub> (> 1000 ppm) and iso-butane (C<sub>4</sub>H<sub>10</sub>) (> 1000 ppm).

Considering that among our proposed DSs, the components are those supported by the default platform (Fig. 2), i.e. their libraries, the code is integrated into the MultispeQ base firmware [17], accessed according to the instructions [19], in particular through the protocols of the PhotosynQ App mobile application [8].

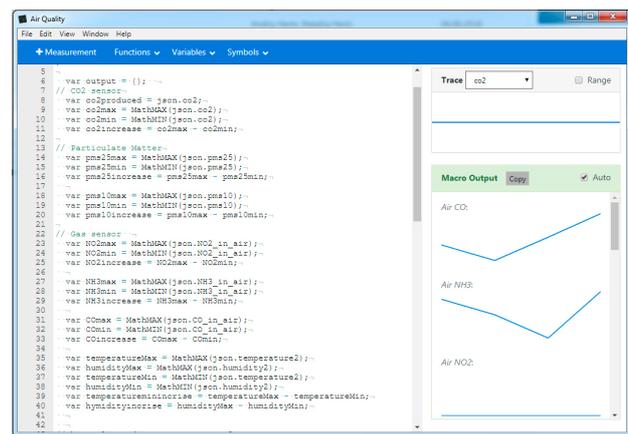
To expand the list of supported DSs, we have made changes to the MultispeQ firmware. In particular, for the Grove Sensor – Multichannel Gas Sensor MiCS-6814 [22], the Seeed Technology Inc. library is built into the definer.h file [17], for Plantower PMS 3003 sensor – PMS Library-1.1.0 [23]. The code (C++) of both these CAs, which contains its own variables and a list of commands, has been integrated into the firmware as separate functions (Fig. 3), which can be called from

anywhere in the sketch, in particular, using the general format of the line instruction (construction) [24]:

```
...
//process single commands
switch (val) {
case hash("get_pms"):
    Serial3.begin(9600);
{
    PMS pms(Serial3);
    PMS::DATA data;
    pms.read(data);
    Serial.println(data.PM_AE_UG_1_0);
    Serial.println(data.PM_AE_UG_2_5);
    Serial.println(data.PM_AE_UG_10_0);
    delay(100);
}
    Serial3.end();
break; }
```



**Fig. 2.** The main components of the prototype for the assessment of the basic physicochemical parameters of the natural environment (A – BME280 [19], C – SenseAir S8 [20], E – HC-06 [21] are supported by the default PhotosynQ platform; B Grove – Multichannel Gas Sensor [22], D PMS 3003 [23] – require additional firmware.



**Fig. 3.** The PhotosynQ Desktop application window for working with macros [8].

To provide communication with the firmware, we used an open access protocol containing instructions written in JSON [25]. It determines which of the DSs will be involved at a particular moment of time. In fact, access to our integrated sensors is provided through a command in the protocol:

```
[ "pms", 0 ], [ "mgs", 0 ]
```

where, “pms”, “mgs” refer to the corresponding variables contained in the firmware.

Therefore, in order to measure all the above-mentioned physicochemical parameters of the medium and to use the DSs mentioned in the work (see Fig. 2), the following construction should be used in the protocol:

```
"environmental_array": [
  [ "co2", 0 ],
  [ "temperature_humidity_pressure2", 0 ],
  [ "pms", 0 ],
  [ "mgs", 0 ]
]
```

The protocols are related to macros [8]. The latter are written in JavaScript and run after the protocol is complete. By providing computations where the primary data is from the JSON CD values, the macros are responsible for outputting the final calculations on the PhotosynQ Desktop personal computer application or, in the case of a mobile phone, or a tablet, in the PhotosynQ App. Actually a macro for calculating the maximum and minimum concentration of CO<sub>2</sub>, the content of particulate matter (particulate matter, (PM)) in the atmosphere, in PhotosynQ Desktop application (Fig. 3) [8] has the following form:

```
var output = {};
// CO2 sensor
var co2produced = json.co2;
var co2max = MathMAX(json.co2);
var co2min = MathMIN(json.co2);
var co2increase = co2max - co2min;

// Particulate Matter
var pms25max = MathMAX(json.pms25);
var pms25min = MathMIN(json.pms25);
var pms25increase = pms25max - pms25min;

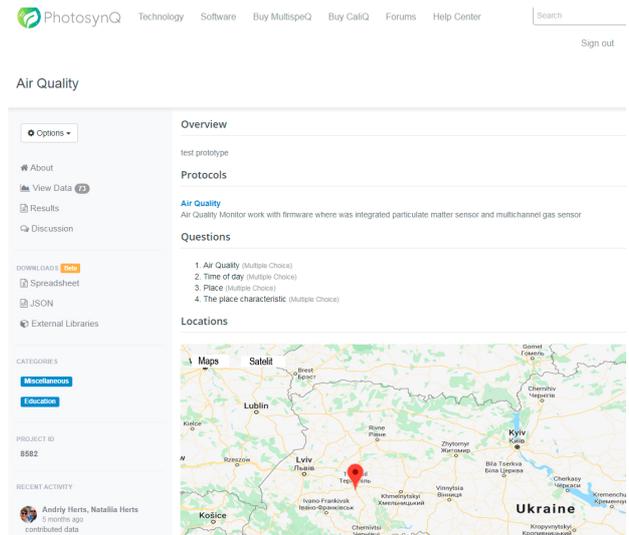
//Show value and name in output CO2
output["co2.array"] = json.co2;
output["CO2 increase"] = co2increase;
output["Max CO2"] = co2max;
output["MinCO2"] = co2min;

//Show value and name in output
// Particulate Matter output
output ["PM2.5"] = json.pms25;
output["PM2.5 increase"] = pms25increase;
output["Max PM2.5"] = pms25max;
output["Min PM2.5"] = pms25min;
```

When using only PhotosynQ Desktop software for writing macros and protocols, the platform’s web interface (Fig. 4) is a place where authorized users can create their own projects [8], view, analyze data, search for projects of other users and, if necessary, join them. Actually, the project acts as a kind of data organization [8].

The project we created with the name “Air Quality” is based on the protocol of the same name (protocol ID 1492), which was described above and contains, in addition to the description, a series of questions that must be answered by each project participant before directly performing the measurement.

This approach provides the necessary structuring, categorization of the obtained data and facilitates their further processing, including statistical data.



**Fig. 4.** Web-based Air Quality project interface on the PhotosynQ platform [8].

In order to participate directly in the project (subject to the creation of the above-mentioned prototype), Air Quality, and make measurements, it is necessary to: 1) ensure communication of the prototype with PhotosynQ App or PhotosynQ Desktop; 2) choose from the list of projects needed; 3) answer all the questions listed in the project; 4) take measurements. After the measurement is completed, it is possible to view its results directly on the screen of a mobile phone, tablet (Fig. 5) or the monitor of a personal computer when using the desktop version of the application. After evaluating the quality of the data, it is confirmed or canceled to send the results directly to the website.

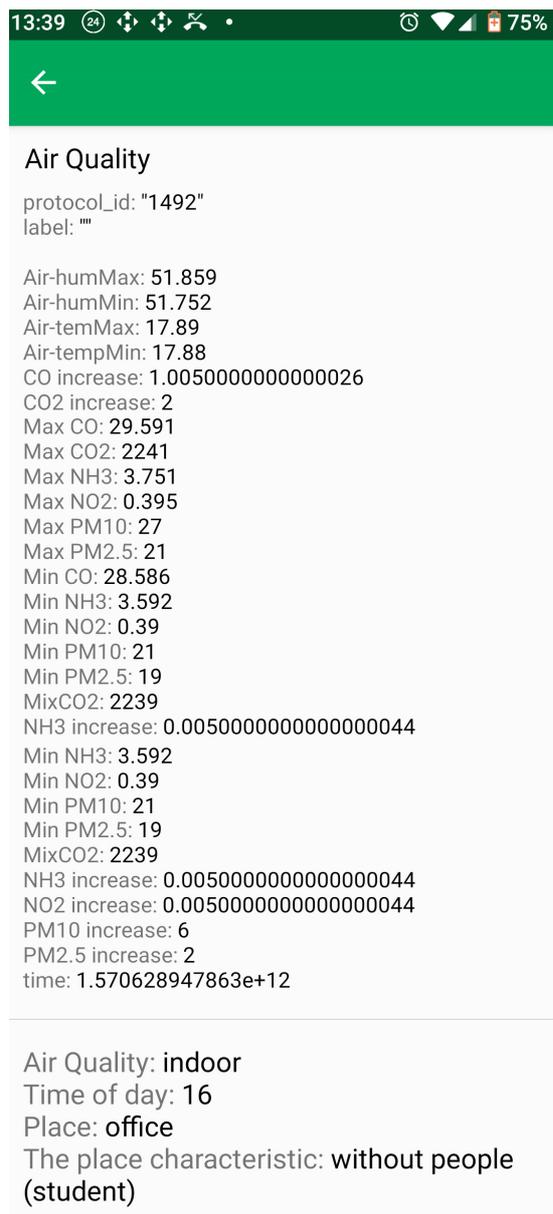
The platform has a whole set of tools for statistical analysis (t-test, chi-square test, ANOVA) and data output. If necessary, for additional statistical processing of the obtained data, in our opinion, it is advisable to use the possibilities of R software environment for statistical calculations, analysis and data representation in graphical form. Providing that there are several ways to access the source data and there is a possibility of downloading them in .csv, .xlsx, json formats, the choice of priority method becomes subjective.

In our opinion, the ability to import datasets directly from a project in Python or R is a very promising solution. In particular, PhotosynQ R (PhotosynQ R package) [7] allows to access our project data by its ID number and generate the required data frame from the R environment.

To generate a data frame file containing the Air Quality project data in RStudio, an integrated development environment (IDE) for R, we executed the recommended [7, 8] set of commands:

```
PhotosynQ::login("login@domain.com")
ID <- 8582
df <- PhotosynQ::getProject(ID)
```

The procedure described in [9] for accessing RStudio software to our aforementioned project with ID number 8582 was successful. For RStudio Cloud, it was similar and did not require the installation of additional libraries.



**Fig. 5.** Presentation of measurement results of the Air Quality project in PhotosynQ App.

Having access, after installation in the Jupyter Notebook environment, which among Data Science professionals is de facto standard for rapid prototyping and analysis of data, PhotosynQ-Python.git from GitHub, to the Web site API [7], each authorized member receives access not only to project data, but also they receive the ability to extend the boundaries of cooperation and beyond.

## 4 Conclusions and prospects for further research

Currently, hardware connectivity and “communication” with most cloud services is happening through an open API. Ready templates for data connection and processing are offered for this purpose. The PhotosynQ platform provides a data access API, but there is no real-time data visualization and a number of other APIs offered by the

services, including ThingSpeak, Ubidot [26, 27]. At the same time, there is an opportunity to display general information about the projects, results and location of the research, etc. This seems logical given that the platform we are researching is primarily aimed at researchers, educators, farmers and citizens who collect, analyze, discuss and exchange information received through portable devices. Considering that the main landmark of PhotosynQ is the collection of data in a well-defined procedure, the user determines the time, place and duration of the process.

The existing firmware contains code that describes the procedure for connecting to it Desktop or mobile application. After receiving information from the device, the commands (procedures) are sent back in JSON format, as they already contain parameters that will determine and define the work of the DSs, the sequence, duration of their work, etc. Therefore, unlike the services mentioned above, which interrogate connected CDs sequentially with a certain user-defined and periodic frequency, the MulispeQ firmware [17], designed to interact with PhotosynQ, is organized so that the user determines the data acquisition procedure, the number of DSs involved at a given time, the sequence and duration of their work, etc. The web platform itself is a place for data collection, analysis and exchange and is not inherently an IaaS or PaaS platform. At the same time, the platform has a whole arsenal of tools for statistical analysis (t-test, chi-square test, ANOVA) and data output.

Therefore, the available firmware used by the PhotosynQ platform to communicate with the fluorometer allows, without further modification, together with the Teensy 3.2 and DSs to obtain, if necessary, a portable measuring device. Support by the BME280, K30 DSs firmware [9] to the color sensor TCS34715 [9] will allow you to evaluate not only temperature, relative humidity, atmospheric pressure, CO2 content, but also the level of illumination in the suites and the spectral distribution in the RGB range. Support for MDX MLX90615 [9] will provide a reading of the surface temperature of the study objects. Available code for the MAG3110FCR1 encoder [9] will allow you to work with a magnetometer and the like.

The suggested approach of inserting additional code into the firmware and successfully plugging in additional sensors, in particular Grove - Multichannel Gas Sensor MiCS-6814, “PMS 3003” enables researchers to tailor the firmware to their own needs, experimenter’s needs, or training process, in particular. By inserting the encoder code into the popular kits Arduino UNO Kit or their counterparts from DFRobot, Elecrow, etc., we will get a tool that organizes the communication of a dozen marketed sensors with IoT-oriented PhotosynQ cloud platform, and the similar ones which will greatly enhance the capacity of not only researchers, teachers but also students.

We thank Greg Austic (Our Sci LCC member/owner) for technical assistance and new opportunity, Sebastian Kuhlger (PhotosynQ LLC) and PhotosynQ team for their cooperation.

## References

1. H. Gottipati, 5 Technology Trends That Will Shape 2017 and Beyond – AI, AR/VR, Analytics, Cloud and Data Science, (2017), <https://medium.com/@harigottipati/5-technology-trends-that-will-shape-2017-and-beyond-ai-ar-vr-analytics-cloud-and-data-science-5787ef529fa9>. Accessed 12 Sep 2019
2. D. Mathew, Why schools should be using cloud based technology? (A Medium Corporation, 2016), <https://highereducation.com/why-schools-should-be-using-cloud-based-technology-ac4da58d7896>. Accessed 11 Oct 2019
3. J. Balasooriya, Cloud Computing Infrastructure for Biological Echo-Systems, in *2010 IEEE 3rd International Conference on Cloud Computing*, Miami, FL, USA, 5-10 July 2010
4. L. Dai, X. Gao, Y. Guo, J. Xiao, Z. Zhang, Bioinformatics clouds for big data manipulation. *Biol Direct.* 7(43) (2012). doi:10.1186/1745-6150-7-43
5. From the lab to the world: solving big problems in agriculture and energy (MSU-DOE Plant Research Laboratory, 2017), <http://prl.natsci.msu.edu/news-events/news/from-the-lab-to-the-world-solving-big-problems-in-agriculture-and-energy/>. Accessed 9 Oct 2019
6. S. Kuhlert, G. Austic et al., MultispeQ Beta: a tool for large-scale plant phenotyping connected to the open PhotosynQ network (Royal Society Open Science, 2016), <http://rsos.royalsocietypublishing.org/content/3/10/160592>. Accessed 14 Oct 2019
7. PhotosynQ LLC (GitHub, 2019), <https://github.com/Photosynq>. Accessed 10 Oct 2019
8. PhotosynQ (PhotosynQ LLC, 2019), <https://photosynq.org/technology> (2019). Accessed 21 Sep 2019
9. D. Bandyopadhyay, J. Sen, Internet of Things: Applications and Challenges in Technology and Standardization. *Wireless Pers Commun.* **58**, 49–69 (2011). doi:10.1007/s11277-011-0288-5
10. I. Houwat, Photosynq's first workshop a success (MSU-DOE Plant Research Laboratory, 2016), <https://prl.natsci.msu.edu/news-events/news/photosynq-first-workshop-a-success/>. Accessed 6 Oct 2019
11. The Experience360 Program Resource Book (2018), <https://www.delval.edu/pdf/e360-program-resource-book.pdf>. Accessed 25 Oct 2019
12. ITU-T Recommendations (ITU's Telecommunication Standardization Sector (ITU-T), 2019), <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11559>. Accessed 19 Oct 2019
13. F. Dressler, A Study of Self-Organization Mechanisms in Ad Hoc and Sensor Networks. *Computer Communications* **31**(13), 3018–3029 (2008)
14. A. Herts, I. Tsidylo, N. Herts, S. Tolmachev, in *Proceedings of the 6th Workshop on Cloud Technologies in Education*, Kryvyi Rih, 21 December 2018
15. What is open hardware? (Opensource.com, 2019), <https://opensource.com/resources/what-open-hardware>. Accessed 10 Oct. 2019
16. Our-sci.net (Our Sci LLC, 2019), <http://our-sci.net>. Accessed 10 Oct. 2019
17. MultispeQ-Firmware (GitHub, 2019), <https://github.com/Photosynq/MultispeQ-Firmware/releases>. Accessed 10 Oct. 2019
18. Teensy USB Development Board (PJRC. Electronic Projects. Components Available WorldWide, 2019), <https://www.pjrc.com/teensy>. Accessed 12 Mar 2019
19. Humidity sensor BME280 (Bosch Sensortec GmbH, 2020), [https://www.bosch-sensortec.com/bst/products/all\\_products/bme280](https://www.bosch-sensortec.com/bst/products/all_products/bme280). Accessed 3 Jan2020
20. Senseair S8 Residential (SENSEAIR, 2019), <https://senseair.com/products/size-counts/s8-residential/>. Accessed 25 Oct 2019
21. Bluetooth Transceiver Module HC-06, [https://wiki.sunfounder.cc/index.php?title=Bluetooth\\_Transceiver\\_Module\\_HC-06](https://wiki.sunfounder.cc/index.php?title=Bluetooth_Transceiver_Module_HC-06). Accessed 9 Oct 2019
22. Grove-Multichannel Gas Sensor (Seeed Technology Co., Ltd., 2018), [http://wiki.seeedstudio.com/Grove-Multichannel\\_Gas\\_Sensor/](http://wiki.seeedstudio.com/Grove-Multichannel_Gas_Sensor/). Accessed 10 Oct 2019
23. Plantower. PM2.5 PMS 3003 (PlanTower, 2015), <http://www.plantower.com/en/content/?107.html>. Accessed 25 Aug 2019
24. D.M. Ritchie, B.W. Kernighan, *The C Programming Language*, 2nd edn (1988), <https://www.oreilly.com/library/view/the-c-programming/9780133086249/>. Accessed 25 Oct 2019
25. *Introducing JSON* (ECMA-international, 2017), <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>. Accessed 23 October 2019
26. IoT Analytics – ThingSpeak Internet of Things (The MathWorks, Inc., 2019), <https://thingspeak.com>. Accessed 21 Mar 2019
27. Data Drives Decisions. IoT platform – Internet of Things (Ubidots, 2020), <https://ubidots.com/>. Accessed 3 Jan 2020