Tvardovskyi, Valentyn; Kravchenko, Olha; Besedina, Svetlana

Article

Design of IoT-solution for monitoring and analysis of the solid waste storage system

Reference: Tvardovskyi, Valentyn/Kravchenko, Olha et. al. (2022). Design of IoT-solution for monitoring and analysis of the solid waste storage system. In: Technology audit and production reserves 2 (2/64), S. 6 - 10.
http://journals.uran.ua/tarp/article/download/255334/252621/587902.
doi:10.15587/2706-5448.2022.255334.

UDC 004.9
DOI: 10.15587/2706-5448.2022.255334
Article type «Reports on Research Projects»

Valentyn Tvardovskyi,
Olha Kravchenko,
Svetlana Besedina

DESIGN OF IOT-SOLUTION FOR MONITORING AND ANALYSIS OF THE SOLID WASTE STORAGE SYSTEM

The object of research is the process of monitoring the filling of garbage tanks of sorted garbage with the help of an automated system based on IoT solution. The paper analyzes existing IoT solutions for monitoring and monitoring the level of fullness of solid waste capacities. The research is important in light of sorting, storage and processing of garbage. The IoT model is proposed for monitoring and analyzing the system of waste recording of solid waste. The IoT solution project involves the creation of a software and hardware for the automated system. The interaction of hardware and software and the implemented root part of the project provide the opportunity to perform the main task – to obtain an assessment of the level of filling capacity for solid waste. Hardware is implemented using a device for analyzing the filling of garbage containers. It is possible to transfer data on the state of the filling of the container. The position of the lid or the same container is taken into account when the container will be inverted or the container lid will be raised. The device itself will be located on the lid on the inside of the container. Also, the device is equipped with a container for burning to prevent potential accidents and emissions into an atmosphere of harmful waste. Software implementation contains the level of programming of tasks for sensors IoT solutions and creates a friendly web-oriented interface. Software implementation is performed on the basis of client-server architecture. To operate the end points of the interaction of local and intermediate servers, the architectural style of REST with the transmission of information in the form of JSON is used. In order to administer an intermediate server, the client part is written on HTML and JavaScript. The application of the IoT solution allows to control the level of filling of containers, optimize routes of garbage trucks, which, in turn, reduces the cost of export of waste for regional operators. The scheme of information exchange in the system of garbage saving is constructed.

Keywords: *client-server architecture, system drafting, monitoring of the filling of garbage tanks, hardware, information exchange scheme.*

Received date: 15.12.2021
Accepted date: 13.01.2022
Published date: 20.04.2022

How to cite
Tvardovskyi, V., Kravchenko, O., Besedina, S. (2022). Design of IoT-solution for monitoring and analysis of the solid waste storage system. Technology Audit and Production Reserves, 2 (2 (64)), 6–10. doi: http://doi.org/10.15587/2706-5448.2022.255334

1. Introduction

Timely cleaning of garbage containers near our homes is an urgent task that needs to be addressed. Analysis of existing solutions shows that there are automated solutions CleanFLEX from ECUBE LABS [1], SmartCity WSens from Binology [2] and BrighterBins from the company of the same name [3]. In the age of the Internet of Things, many platforms based on microcontrollers or microcomputers are designed to serve IoT programs and systems. The microcomputer has an interface that can be accessed by connecting it to a monitor. It is common practice for a microcomputer to have an operating system (OS). The microcontroller does not have an interface, so the program must be recorded on a computer and downloaded to the board. It can save and run only one program at a time, but can be reprogrammed several times. Most often, users get three technologies when searching for information about IoT software and systems: Arduino, Raspberry Pi and Microbit. Analysis of existing IoT solutions for control and monitoring of the level of solid waste containers and the condition of containers near our homes allows to talk about the relevance of this topic.

Therefore, *the object of research* is the process of monitoring the filling of garbage cans of sorted garbage using an automated system based on IoT solutions. *The purpose of the study* is to build a draft IoT solution for monitoring and analysis of solid waste storage system.

2. Research methodology

Fig. 1 shows the growing popularity of web search platforms for IoT applications and systems over the past 5 years around the world, based on microcontrollers or microcomputers [4]. Taking the data from the chart as a basis, it is possible to conclude that the undisputed leader in popularity is the Arduino, followed by the Raspberry Pi and, compared to the previous two options, Microbit completely loses in popularity of information about it.

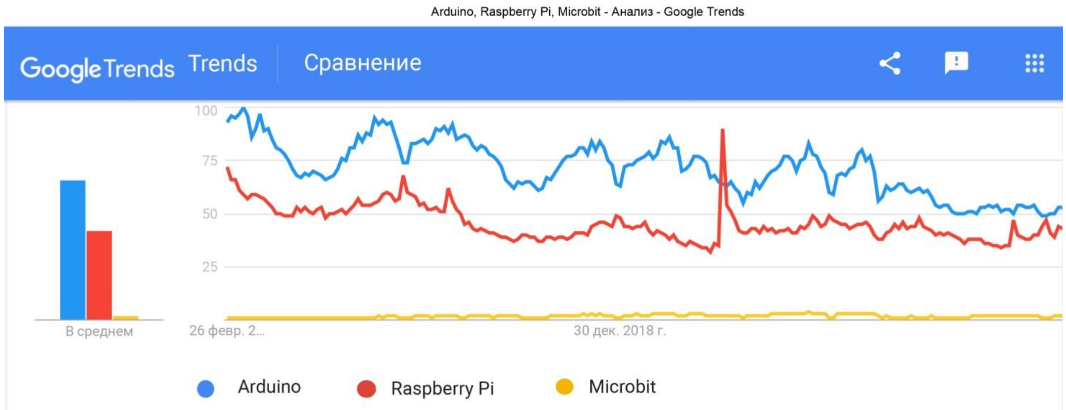


Fig. 1. The dynamics of growing popularity in web search platforms based on microcontrollers or microcomputers designed to service programs and IoT systems [4]

Despite the continued dominance of Arduino in search queries – in the period from 23 to 29 June 2021 there was a sharp explosion in the popularity of information retrieval about the Raspberry Pi. Every year the difference in popularity of the two main competitors decreases. The Raspberry Pi is a microcomputer designed for software development. Powerful processing and networking capabilities make it suitable for video/audio processing and as a server. For a fair comparison with all platforms in this paper, the specification comparison between the Raspberry Pi models adapts the same structure as the specification comparison from the Arduino microcontroller. However, features such as analog I/O (F4) and UART port (F5) have been omitted and replaced by SoC and network. Analog I/O is omitted because the Raspberry Pi does not require ADC/DAC ports, as these two units are available inside the SoC. As for the UART port, the digital I/O for the Raspberry Pi has a lower-level UART interface that is accessible via the USB port. Fig. 2 shows the hardware of the IoT solution being designed.

The Raspberry Pi will be taken as a basis, as each device must have a local server that will collect information from sensors, process this information for further transmission in a simplified unambiguous form [5].

Among the distance sensors, the ultrasonic distance sensor HC-SR04 (China) will be used to assess the fullness of the container. This sensor sends a signal in a certain range, which significantly affects the accuracy of checking the occupancy of the area. The gyroscope-accelerometer GY-521 (China) will be used to check the location of the container in relation to the normal state. It is possible to check the temperature and humidity of the device using the digital sensor AM2320 521 (China). This will help both as a test of the container's combustion and, if necessary, to adjust the ultrasonic distance sensor, as the temperature of this distance sensor may vary by a few centimeters depending on the temperature. A MQ-2 gas sensor (China) will be used to check for smoke. Its range of measured gases will be sufficient for premature inspection of the container before it is fully ignited and will prevent, by sending a message to the server, the release of hazardous and harmful gases into the atmosphere. Data will need to be transmitted over the network, so NB-IoT technology is chosen for this purpose. This technology is evolving rapidly and is easy and cheap to integrate for each device, even if it is one device per quarter (i. e. one container). Simplicity in debugging is achieved due to the minimal need to install a base station. For example, most Ukrainian providers provide their services using NB-IoT.

Fig. 2. Hardware support of IoT solutions for monitoring and analysis of solid waste disposal system: *a* – Raspberry Pi 4 Model B; *b* – HC-SR04; *c* – GY-521; *d* – AM2320; *e* – MQ-2

The server is configured on the one hand using the Java programming language, namely the Spring Boot framework. Because it will help to easily, quickly and efficiently set up a local Tomcat server. Let's use the Python programming language, namely the Flask framework, to «pick up» another local server that will read data from sensors and transmit it to the Tomcat server. It is also necessary to prepare an intermediate server, which will also use the Spring Boot framework. An intermediate server will help optimize the information coming from a large number of devices. Since there can be several garbage containers in one place, the information is sent in a composite form. The REST architectural style with JSON information transfer was used to design the endpoints of the local and intermediate servers. To be able to administer the intermediate server, the client part will be written using HTML and JavaScript.

Fig. 3 shows the scheme of interaction between all components of the project.

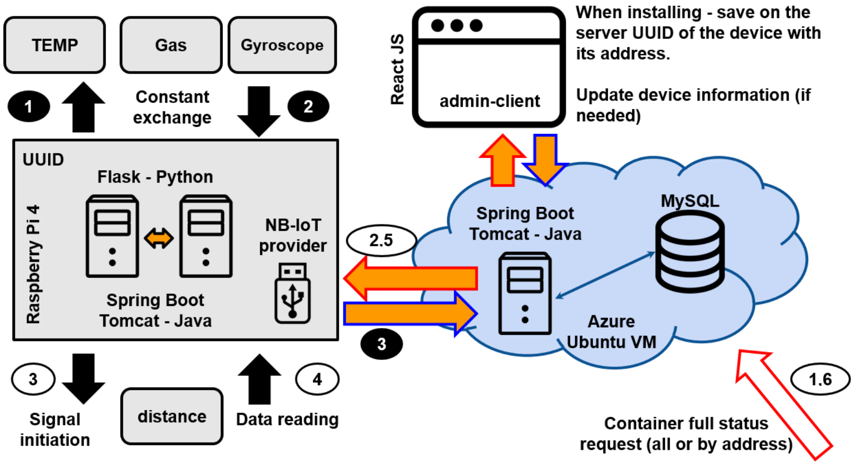


Fig. 3. Scheme of interaction of IoT solution components for monitoring and analysis of solid waste storage system

According to the scheme in Fig. 3, the exchange of information is as follows:

1. A company that needs container fill status information makes an HTTP GET request with an address, using an intermediate server API. Or, as an alternative to integration – the company can get the necessary information on the site, or download a CSV file from there.

2. The intermediate server receives the request and searches the database for stored containers at the specified address. It then makes an HTTP GET request for each container to update the fill status information of each one.

3. A local Tomcat server running on a Raspberry Pi receives a request from an intermediate server and requests a local Flask server. In turn, the local Flask server, which is directly responsible for reading data from the sensors – initiates the signal in HC-SR04.

4. The ultrasonic distance sensor HC-SR04 reads signals returned from objects, processes these signals and transmits information to the Flask server.

5. The Flask server returns information to the Tomcat server. And the Tomcat server, accordingly, returns the received information to the intermediate server.

6. The intermediate server receives a response to the query status of each container, stores updated information in the database and generates a response to the company's request.

To ensure reliability, the container is tested by three more sensors – temperature sensor, gas sensor and gyroscope [6, 7]. The logic of their interaction is as follows:

1. Every minute, the Tomcat server queries the Flask server in turn for each of the above sensors. In turn, the local Flask server initiates the reading of data from the sensors.

2. Sensors receive and process information, then return it to the Flask server. The Flask server, accordingly, transmits the received information to the Tomcat server.

3. The Tomcat server analyzes the data received from the sensors and, if necessary (for example, when information from the gyroscope indicates that the container has been inverted) sends emergency information to the intermediate server. The intermediate server, in turn, displays a message about the status of the container on the site, as well as sends an e-mail or message in the messenger to the responsible person.

3. Research results and discussion

Java 8 OpenJDK is installed on the Raspberry Pi to develop the software part of the project, and Python has already been installed.

Additionally, Python packages for Flask server are downloaded. Raspberry Pi OS is used as the operating system of the microcomputer [8]. First of all, the data reading from the sensors must be configured. To do this, the project uses Python. The distance.py file has been created, which imports the necessary libraries for interaction with microcomputer contacts and for raising the local Flask server [9, 10]. The intermediate server makes a request for this endpoint, and from there the request is sent locally to the Flask server. Each device has its own unique UUID to identify the device and it is also used as part of the endpoint path (Fig. 4). The server part of the Raspberry Pi is ready to work, so it is necessary to develop an intermediate server – the cloud, which is responsible for interaction with project clients, as well as the business logic of the project components.

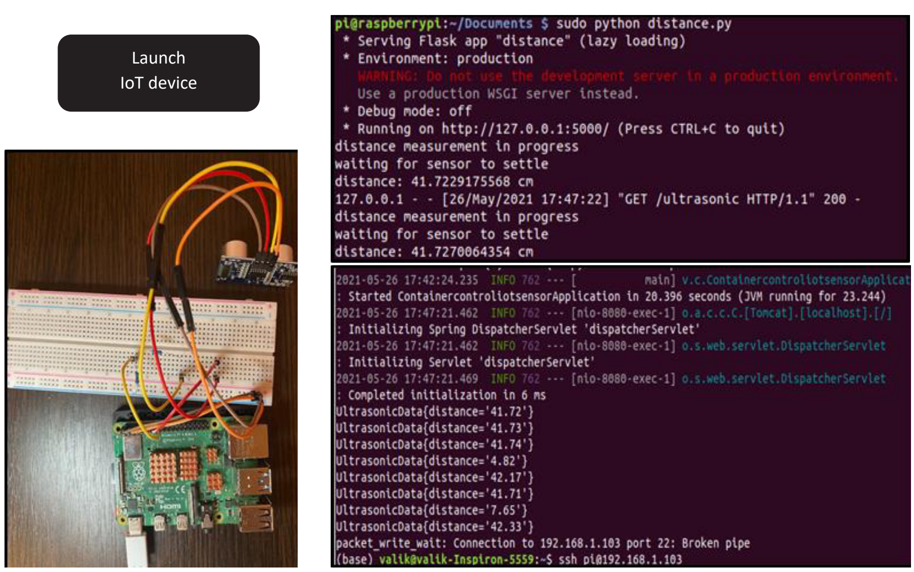


Fig. 4. Prototype of the draft IoT solution for monitoring and analysis of solid waste storage system

In the process of creating a prototype, only ready-made inexpensive, but well-known and long-tested for efficiency and stability, chips and sensors are used. Therefore, any part of the device can be easily replaced or repaired by any engineer, or even «advanced» user, based on information about each sensor from this device on the Internet.

Spring Boot, Tomcat, Maven and, accordingly, Java are also used as a server in the cloud. MySQL database is used to store information on the status of containers, etc. [11]. To obtain information, it is also possible to use the project site, shown in Fig. 5.

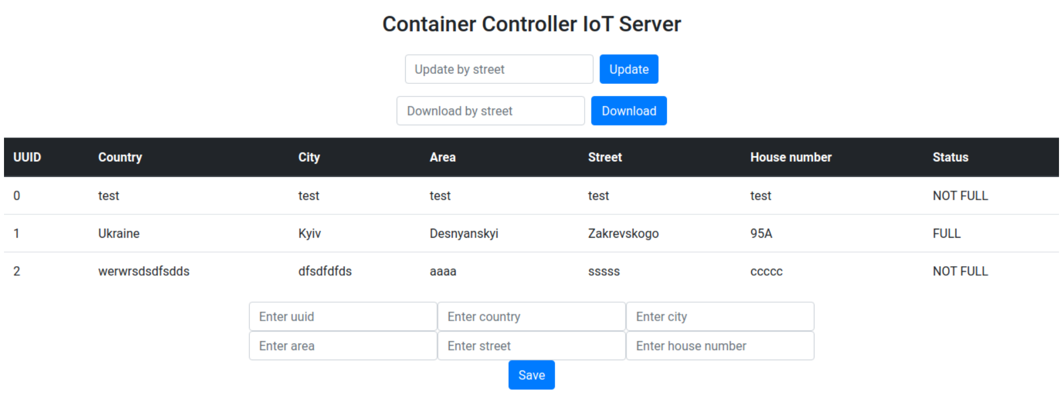
To start the IoT device, connect to the power of the microcomputer. After the operating system starts – connect to Raspberry Pi via ssh and boot local Flask and Tomcat servers. The device is ready for data transfer. Using the website it is possible to get information about the status of containers. Through the API information is transmitted in JSON format, and through the site let's receive visual information with containers with the ability to download CSV file.

The IoT is based on the Raspberry Pi 4 Model B. It has 8 GB of RAM. A set of copper radiators is used to ensure more efficient cooling of the system. The power supply is a 5.1 V 3 A unit with USB Type C interface. The HC-SR04 ultrasonic distance sensor has four contacts. The VCC and GND contacts are used to connect power, and the Trig and Echo are used to send and receive rangefinder signals.

Fig. 6 shows a list of contacts of the microcomputer. The supply voltage of the rangefinder is 5 V, so its VCC contact is connected to pin 2 on the Raspberry Pi. The GND pin, respectively, is connected to pin 6 on the Raspberry Pi. Then the Trig pin is connected to pin 16 on the Raspberry Pi to initiate the signal, but with the Echo pin it is a bit more complicated. The fact is that the Raspberry Pi contacts receive a signal with a voltage of 3.3 V, but the contact Echo rangefinder gives a signal with a voltage of 5 V. The same with the help of resistors 2 kOhm and 1 kOhm voltage was reduced.

Because an ultrasonic distance sensor emits a signal from any object in the path, there is always the possibility that someone will close the sensor with some large object that will not actually occupy the entire volume of the container. For example, these may be foam blocks for transporting the refrigerator. They are high and relatively wide, so they can easily cover the visibility of the sensor. This is a human factor and cannot be completely processed in the case of using open containers. One of the simplest and perhaps most effective solutions would be to focus on the location of the device in the container with a bright sticker or something similar. After all, ordinary garbage containers do not have any means of influencing the placement of garbage (such as ramming or other).

The connection components are ready for further software development. According to the project plan, the device will also be powered by a battery instead of a power supply, as the device will be independent of the location of power sources. The batteries will be replaced, so the device can work without problems for years.



UUID	Country	City	Area	Street	House number	Status
0	test	test	test	test	test	NOT FULL
1	Ukraine	Kyiv	Desnyanskyi	Zakrevskogo	95A	FULL
2	werwrsdsdfsdds	dfsdfdfds	aaaa	sssss	ccccc	NOT FULL

Fig. 5. Main page of the IoT project solution website for monitoring and analysis of solid waste disposal system

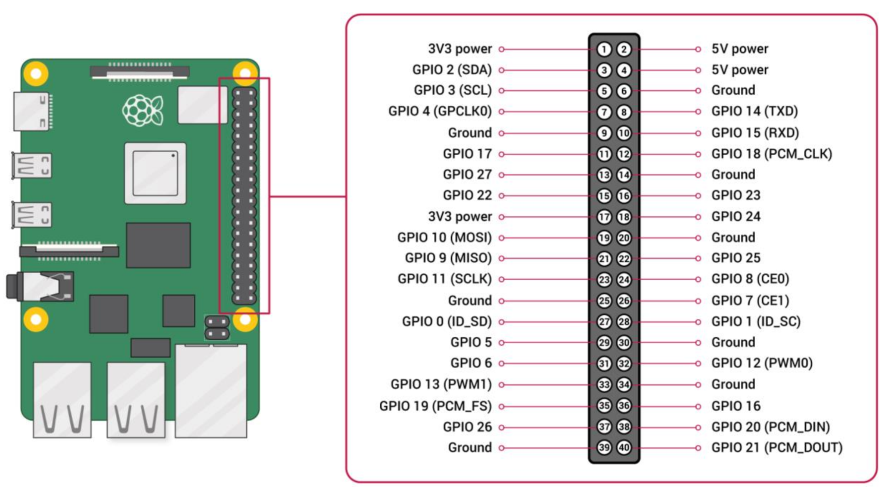


Fig. 6. Contacts of Raspberry Pi 4 Model B

 ISSN 2664-9969

As one of the key issues in the success of the project is the pricing policy, it is worth exploring the possibility of using a simpler, and therefore cheaper, version of the Raspberry Pi microcomputer. Based on the data obtained, the Raspberry Pi model was selected that fully provides the necessary power for the entire device and at the same time will not have redundant capabilities and capacity.

4. Conclusions

The paper analyzes the existing IoT solutions for control and monitoring the level of solid waste containers. Hardware and software for designing similar IoT solutions are also analyzed. As a result, our own IoT device is designed and the root part of its functionality is implemented.

It is possible to monitor the status of the container, and therefore collect statistics and further automate the process of garbage collection by utilities/commercial enterprises with the help of this device based on the data obtained. The ability to monitor the condition of the container leads to a reduction or complete absence of unnecessary departures of garbage trucks, and thus reduce the cost of waste disposal. It also solves the problem that overflowing containers leads to increased unsanitary conditions, the creation of local landfills, access of animals and birds to waste.

The research results can be used in the design of IoT devices with similar functionality, and the designed and developed solution can be completed and supplemented for practical use and for commercial purposes.

References

1. *Ecube Labs, CleanFLEX, the fill-level sensor*. Available at: https://www.ecubelabs.com/ultrasonic-fill-level-sensor/
2. Datchyk urovnia zapolnenyia Smart City Waste Sens. *Binology*. Available at: https://www.binology.ru/sensor
3. Our solution. *BrighterBins*. Available at: https://www.brighterbins.com/the-solution
4. *Dynamyka populiarnosty Arduino, Raspberry, Microbit*. Available at: https://trends.google.com/trends/explore?date=today%205-y&q=Arduino,Raspberry%20Pi,Microbit
5. Qamarina, N., Azwady Jamaludin, A., Azizan, A., Abas, H. (2018). Arduino vs Raspberry Pi vs Micro Bit: Platforms for Fast IoT Systems Prototyping. *Open International Journal of Informatics, 6 (1)*.
6. Raspberry Pi OS. *Raspberrypi*. Available at: https://www.raspberrypi.org/documentation/usage/gpio/
7. Ultrazvukovyi datchyk vidstani HC-SR04. *Arduino*. Available at: https://arduino.ua/prod182-yltrazvykovoi-datchik-rasstoyaniya-hc-sr04
8. Laser Sensor User Manual SR04. *Waveshare*. Available at: https://www.waveshare.com/w/upload/9/93/Laser-Sensor-UserManual.pdf
9. *IR datchyk rasstoianyia Sharp GP2Y0A21YK0F*. Available at: https://www.robostore.com.ua/moduli-i-datchiki/sensory-datchiki-i-moduli/ir-datchik-rasstoyaniya-sharp-gp2y0a21yk0f-10-80-sm/
10. Modul datchyka polumia KY-026 dlia Arduino. *Robostore*. Available at: https://www.robostore.com.ua/ua/modul-datchika-plameni-ky-026-dlya-arduino/
11. How to Choose the Right Language for an IoT Project? (2020). *Scand*. Available at: https://scand.com/company/blog/top-languages-for-iot-development/

Valentyn Tvardovskyi, *Department of Information Systems and Technologies, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, ORCID: https://orcid.org/0000-0001-6048-7023*

✉***Olha Kravchenko,*** *PhD, Associate Professor, Department of Information Systems and Technologies, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, ORCID: https://orcid.org/0000-0002-9669-2579, e-mail: kravchenko.olga0803@gmail.com*

Svetlana Besedina, *PhD, Associate Professor, Department of Information Technologies, The Bohdan Khmelnytsky National University of Cherkasy, Cherkasy, Ukraine, ORCID: https://orcid.org/0000-0002-5391-643X*

✉*Corresponding author*