UDC 004.031.43:004.76

IOT COMPONENTS FOR PRODUCTION QUALITY MONITORING

U.A. VISHNYAKOU, S.H. AL-HAJJ, A.H. AL-MASRI, B.H. SHAYA

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus

Submitted 1 November 2021

Annotation. To automate the creation of IoT systems, design tools are used in the form of IoT platforms. The structure of the stack in the IoT network is considered. The connection of sensors with means of primary processing, including protocols and data structure, is described. A generalized algorithm for creating a network using the IoT plat-form Bluemix from IBM is presented. The forms of the developed user interface are described.

Keywords: IoT networks, IoT platform components, IoT network modeling.

Introduction

With the continued development of large and small networks in infocommunication (WAN, LAN) intended for the Internet of People (IoP), a variety of Internet of Things (IoT) networks are becoming more widespread [1].

The structure of the IoT network includes milk analyzers, gateways-converters, a cloud platform, and mobile devices [2]. The cloud platform rents a server that hosts knowledge and data bases, special software (solver) for processing and making decisions on milk quality, and a farm website. The database of the cloud structure server stores milk quality characteristics, and the knowledge base stores the rules for processing them. The solver outputs deviations from the current milk quality indicators from the standards. The site is used for communication of specialists in milk quality control. Monitoring of milk quality characteristics is implemented from mobile devices of specialists with access to the site components.

The 4th generation LTE network using NB-IoT technology was chosen as the network for transmitting information from dairy farms to the cloud. The review of milk analyzers of both domestic and foreign companies is carried out. A gateway solution for querying milk analyzers and transmitting parameters to the cloud infrastructure is presented [3, 4].

IoT structure

As part of the Internet of Things network, five components can be distinguished [5]: sensors (devices) and their hardware, software for sensor management, communication tools, cloud platform and mobile applications. Let's consider the purpose of each of these components [2].

Devices act as an interface between the physical and digital worlds. They are the first layer of the IoT technology stack. Only one sensor may be needed for simple data collection. For more complex data, collection, it may need a computer that contains many sensors, a processor, local storage, a gateway, etc. At this level of the IoT technology stack, it is important to understand such parameters of equipment as cost, size, ease of deployment, reliability, useful life, service, etc.

Device software is the second layer of the IoT technology stack. It is a component that turns a device's hardware into a «smart device». Device software includes the concept of «software-defined hardware», which means that a particular hardware de-vice can serve multiple applications depending on the firmware it runs on.

The device software allows communication with the cloud or other local devices. You can perform analytics in real time, collect data from device sensors and monitor parameters. The device software layer consists of two components: the operating sys-tem and the device applications. If the device is simple, the OS may not be used. The application can analyze the data from the sensors, comparing them with the boundary values. If the data exceeds the permissible, it is transmitted to the cloud for monitoring by the operator.

Communications − the third level of the IoT stack, includes both physical net-works and the protocols that will be used. The implementation of the communication layer can be found in the hardware and software of the device. But in terms of the conceptual model, you can keep communication as a separate layer to facilitate discussion during development.

Choosing the right communication mechanisms is an important part of an IoT product strategy. It will determine not only how data is transferred to and from the cloud (using different subnets and protocols: Wi-Fi, 4G, LoRA, etc.), but also how communication with other devices is organized.

The cloud platform (the fourth layer of the IoT stack) is the basis of the IoT network (project), which provides the infrastructure that supports all components: de-vice management, data collection and management, data analytics, cloud interfaces, in-formation security. Smart devices will transmit information to the cloud. You need to be aware of the type and amount of data that will be collected daily, monthly and annually.

Analytics refers to the ability to process data, find patterns, make predictions, integrate machine learning, and so on. The ability to find information from data makes a solution useful. Analytics can be as simple as aggregating and displaying data, or as complex as using machine learning or artificial intelligence. Cloud interfaces allow clients and managers to either interact with devices or exchange data. You may need separate applications for desktops, mobile devices, and for different categories of users.

Algorithm for connecting sensors with primary processing

For simulated IoT monitoring of parameters, we use a scheme for reading readings of measured physical quantities (indicators of product quality or environmental parameters). Then they are preprocessed and the application is sent to the client via the IoT platform. The final stage is displaying data on the client side. The generalized net-work design algorithm is represented by three steps:

1. Select a sensor or a set of sensors from which we will receive the measured data, and a method for processing the received data.

2. Determining how we will communicate with sensors, determine the amount of data and understand how we will build interaction.

3. Find a suitable client for our network and describe the work with him.

For the sensor and pre-processor, we will choose the SensorTag 2 circuit from Texas Instruments [6]. Another option is Arduino with BLE-shield (Bluetooth low energy, low power) or BLEduino.

Inside the CC2650 chip, the SensorTag 2 core, is a real-time operating system (TI-RTOS), which together with the BLE stack provides reliable control of three different microcontrollers:

1. The core of the first microcontroller is Cortex-M3 (it usually runs the custom application we have written).

2. The core of the second - Cortex-M0 (responsible for the physical layer, radio communication).

3. A separate controller for sensors (helps to quickly receive data from them).

Android phones with BLE stack support are widespread, we will use it as a hub and gateway on the way to the cloud.

Generalized algorithm for creating a network by means of the IoT platform. Consider an IoT network modeling algorithm, which is divided into two parts. First, we organize the transfer of information from the sensors using a smartphone (three levels of the IoT stack), then we connect the cloud platform and implement the remaining two levels of the IoT stack.

To send the data received to the gateway (smartphone) via the BLE protocol to the Internet, we use the MQTT protocol and the JSON data transfer format.

MQTT (Message Queue Telemetry Transport) is a simplified network layer protocol for exchanging messages between devices, it runs on top of the TCP/IP stack and is designed to connect sensors, microcomputers, smartphones, tablets. MQTT is a publisher/subscriber. A publisher (devices of type publishers) sends a message, which is published in a centralized service (a message broker), and a subscriber (devices of type subscriber) receives a message from the broker.

JSON (JavaScript Object Notation) is a textual data interchange format based on JavaScript and used with this particular language. The format is considered language independent and can be used with any programming language. For this, there is a ready-made code for creating and processing data in JSON format.

JSON text is (encoded) one of two structures:

− a set of key-value pairs. In various languages, this is implemented as an object, record, structure, dictionary, hash table, keyed list, or associative array. The key can only be a string (case-sensitive: names with letters in different cases are considered different), the value can be any form.

− an ordered set of values. In many languages, this is implemented as an array, vector, list, or sequence.

These are universal data structures: as a rule, any modern programming language supports them in one form or another. They formed the basis for JSON, as it is used to exchange data between different programming languages.

A common implementation of the MQTT protocol is the Paho MQTT library, which is implemented for common programming languages: C/C++, Java, JavaScript, Python, etc. Let's consider the algorithm of communication between the client and the cloud.

1. Import the Paho MQTT library and the classes we need to work with the MQTT protocol.
2. Indicate the address of the cloud.
3. Set the number of the standard port of the broker of the cloud platform of the MQTT protocol.
4. Send data to the cloud platform.
5. The hub/gateway device (android phone) generates MQTT packets and transmits them to the cloud for storage and processing.
6. From the cloud, the hub can receive commands for device control or for the gateway.

Then we will create our web service to receive and display the readings of our sensors. To do this, we will use the cloud platform from IBM − Bluemix [7], which is needed to implement storage services, analytical processing and visualization of data received from SensorTag and pumped through an android phone. Bluemix is a PaaS (Platform as a Service) open source cloud offering based on the Cloud Foundry open source project. The platform is designed for application development and hosting, and it also simplifies infrastructure management.

Bluemix platform using

Building a cloud platform app. In Bluemix terminology, an application is any generated code (source code or executable binaries) that must be run or referenced at runtime. Mobile apps run outside of the Bluemix environment and use the services provided by the apps. In the case of web applications, an application is code uploaded to the Bluemix platform for hosting purposes. In addition, the platform is capable of hosting the application code that we want to run on an internal server in a container-based environment [7].

A service is code that runs on the Bluemix platform and offers specific functionality that applications can use. This can be a ready-made service used directly, such as push notifications for mobile apps or elastic caching for a web app. You can create your own services ranging from simple utility functions to complex business logic.

There are three steps to using services in Bluemix:

1. Tell the Bluemix platform that we need a new instance of the service and specify which particular application will use this new instance.
2. Bluemix automatically initializes a new instance of this service and associates it with the application.
3. The application interacts with the service.

Service bundles are collections of APIs used in specific areas. For example, the Mobile Services package includes MobileData, Cloud Code, Push, and Mobile Application Management services. The available services and runtimes are listed in the Bluemix catalog. In addition, you can register your own services.

In the platform, we will choose Node-RED because of its convenience and ease of configuration. It has a convenient graphical programming interface consisting of JS blocks, which can be described by loading a JSON file. You can also use some Node.js.

The data from the MQTT package is sent to the broker as part of the IoT Foundation service. A data subscriber is a Node-RED application that allows you to manipulate data using simple visual aids. Many primitive processing units (nodes) are JavaScript applications connected to each other by streams of data.

User interface forms

The authors have developed a number of screen forms for working with users through applications. They can be grouped as follows.

1. Input.
2. Special.
3. Profile.

The user's work begins with the choice of language (forms input, Fig. 1). Then the user is authenticated with the ability to input the login and password (input form, Fig. 2).

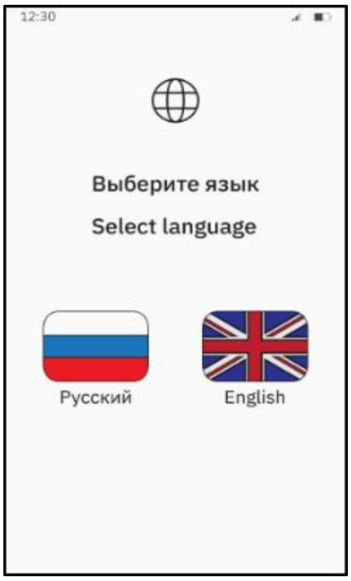


Fig. 1. Select language

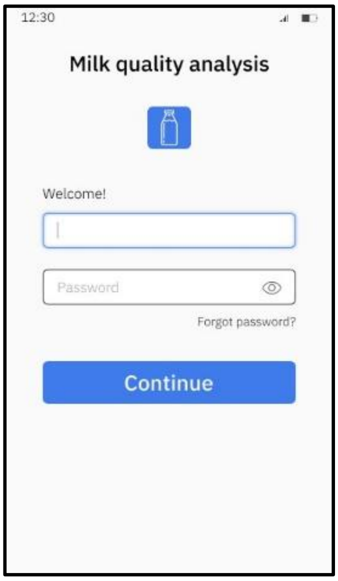


Fig. 2. Input login, password

In a special mode, the user can view product quality parameters from some farm and for a certain period of time (Fig. 3). In case of the output of some parameter foreign values, the user receives a notification (Fig. 4).

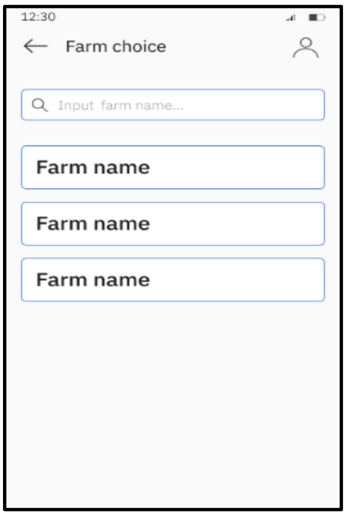
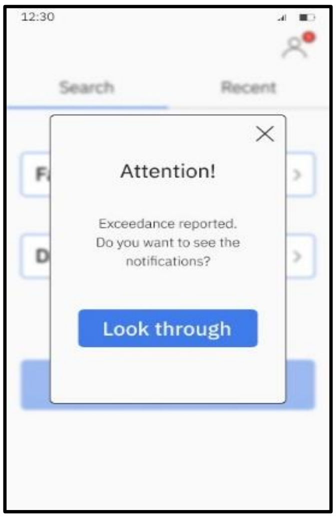


Fig. 3. Input firm for control



Exceedance reported.
Do you want to see the notifications?

Fig. 4. Some parameter foreign values

Profile forms allow user to change his account and language.

Conclusion

The article considers the structure of the IoT network for monitoring milk quality includes analyzers, gateways-converters, cloud platform, and mobile devices. The connection of sensors with means of primary processing, including protocols and data structure is described. A generalized algorithm for creating a network using the IoT platform Bluemix from IBM is presented. The authors have developed a number of screen forms for working with users through applications.

References

1. Roslyakov A.V. Internet of things: textbook manual. Samara: Pgutii, 2015.
2. Visniakou U.A. [et al.] // SA&AI. 2021. № 1. P. 39−44.
3. IoT Platforms. [Electronic resource]. URL: http://www.tadviser.ru/index.php.
4. Rentiuk V. // Control Engineering. Россия. 2018. № 3(75). P. 82−87.
5. Daniel Elizalde. The 5 Layers of the IoT Technology Stack. [Electronic resource]. URL: https://danielelizalde.com/iot-primer/.
6. How to create an IoT for yourself. Learning to do the Internet of things on Android and hardcore hardware. [Electronic resource]. URL: https://xakep.ru/2016/04/28/iot-android-sensortag/.
7. Platform description Bluemix. [Electronic resource]. URL: https://searchcloudcomputing.techtarget.com/definition/IBM-Bluemix.