Industrial internet of things in production of cooked smoked sausages

A M Popov, D M Popov, M Mukhim-zade and N A Wegner

Kemeroovo state university, Krasnaya str., 6, Kemeroovo, 650000, Russian Federation

E-mail: pdm2005@rambler.ru

Abstract. The producers of meat products and semi-finished products often use the equipment of the 90s. Such equipment does not possess modern means of transmitting diagnostic data and information about the current state wirelessly, and requires periodic monitoring during a shift. Lack of timely diagnostic data can lead to work of equipment failures and unforeseen downtime. At the same time, the development of technical means of communication allows connecting industrial equipment to the Internet and receiving data on the current status of the equipment remotely online. In industrial production, this technology is called the industrial Internet of things. The article discusses the use of the ESP32 microcontroller, the Message Queuing Telemetry Transport network protocol and the corresponding server and client software for connecting the MULTIVAC R5200 packaging machine to the industrial Internet of things. It is shown that the use of the industrial Internet of things when packing cooked-smoked sausages can reduce the time of maintenance of the packaging machine and the consumption of spare parts, and reduce the amount of marriage. In the future, the spread of such technology to the equipment of the entire enterprise allows solving the problems of collecting a large amount of data on the current state of equipment, conducting analysis and making managerial decisions.

1. Introduction
A feature of modern industrial production of meat products and semi-finished products is the use of automation technologies, which can not be called relevant [1]. Often, 30-year-old manufacturing equipment runs on serial RS485 interfaces rather than modern wireless mesh architecture. At the same time, the world is witnessing a rapid growth in the introduction of Internet of Things (IoT) devices into industrial production.

IoT is a technology that integrates devices into a computer network and allows collecting, analyzing, processing and transmitting data to other objects using software, applications and technical devices. For the most part, devices operate without human intervention, although people can interact with them: configure, give instructions, or provide access to data. Similar systems work in real time. The introduction of the IoT appeared due to the widespread adoption of the Internet, smartphones and wireless networks. According to forecasts in 2020, from 2.4 to about 4.9 trillion dollars will be involved in the IoT [2, 3].

The largest and fastest growing segment of the Internet of things is the industrial Internet of things (IIoT). This segment includes hardware and software tools for monitoring physical devices. The characteristics of the IIoT segment include the need to provide turnkey solutions to the technology system in real time or almost real time. This means that in everything related to the production hall,
the main parameter for the IoT will be response time. In addition, downtime and safety will play a critical role. This implies the need for high performance computer hardware and, probably, the availability of private cloud networks and data warehousing. The Industrial IoT is one of the fastest growing segments in this market [4, 5].

Therefore, according to our opinion it is important to develop a methodology for implementing the IoT technology in the industrial production of meat products and semi-finished products. The method of connecting industrial equipment to the Internet of things will be considered using the example of a vacuum thermoforming packaging machine of deep drawing MULTIVAC R5200 in the production line of cooked smoked sausages. The machine was made in Germany in 1990.

2. The purpose of the study

The purpose of the research was to study the possibility of implementing the IIoT technology in industrial production, using the example of one piece of equipment. The goal was to determine the effectiveness of the technologies used to monitor the operation of equipment in real time, assess implementation costs and estimated economic and other effects and develop a methodology for connecting old equipment to the Internet of things.

3. Materials and methods

3.1. Microcontroller communication with packaging machine

The first objective of the study was to obtain information about the state of the packaging machine during the technological cycle. To do this, it was necessary to organize the output from sensors and measuring instruments to the microcontroller. The names of sensors and measuring instruments used to monitor the machine via the Internet are shown in table 1.

| №  | Names                                      | Amount |
|----|--------------------------------------------|--------|
| 1  | Regulator relay PID TRM101-RR              | 2      |
| 2  | Three-phase shield multimeter Omix P99-MLY-3-0,5-RS485 | 1      |
| 3  | Current transformer 250A/5A, CLASS 0.5, 6VA | 3      |
| 4  | Inductive Sensor NI 10-G 19-AP9            | 3      |
| 5  | Limit switch                               | 3      |
| 6  | Reed switch                                | 4      |

To study the operation of the packaging machine and obtain a cyclogram of its work, the Saleae Logic analyzer and the corresponding software were used. The sensors of the machine were connected via optocouplers PC817. The resulting sequence diagram of the packaging machine is shown in Figure 1. The states are indicated in the figure: S39 - the tool is open; S40 - close forming tool; S41 - close the welding tool; S47 - start feed motor; S43 - stop feed motor; S44 - control of the printed label. The packaging machine has the ability to regulate in the vacuuming time, the operation of other mechanisms is rigidly set by the cam mechanism. Thus, by connecting the outputs from the sensors to the microcontroller, it is possible to get the state of the machine at any time.

The ESP32 microcontroller is used to monitor the condition of the packaging machine and transfer the received data to the Internet. ESP32 was created and developed by Espressif Systems, a Chinese company based in Shanghai, and manufactured by TSMC using a 40 nm process technology. The ESP32 microcontroller uses two Tensilica Xtensa LX6 cores. This microcontroller has low power consumption. The difference character the ESP32 system is the built-in Wi-Fi controller that we need to connect to the edge router in the workshop. ESP32 programming is carried out using the official
development environment Espressif IoT Development Framework (ESP-IDF), it is possible to use the Arduino IDE [6, 7].

![Sequence a cycle diagram of Packing machine](image)

**Figure 1.** Sequence a cycle diagram of Packing machine

The TRM101 temperature controller has an RS485 interface and the Owen protocol is used for communication. The Omix multimeter also works on the RS 485 interface, but using the Modbus protocol. Therefore, to communicate with these devices, two serial ports of the ESP32 microcontroller had to be allocated. To obtain the sequence diagram, five ESP32 Input/Output pins were allocated. They decided not to use information about the management of printed labeling. In addition, four more pins were used to connect reed switches from various protective guards of the machine.

To process such a large amount of information on the ESP32 microcontroller, the FreeRTOS real-time operating system was used. The processing of data from each group of contacts and individual devices was put in a separate task of this operating system. Sending the received data via the WiFi server is also programmed into the FreeRTOS task.

### 3.2. Microcontroller communication with a cloud service and data presentation on user devices

There are many protocols for exchanging data over a network. For the purposes of the IIoT, the protocol has a number of requirements:

- Ease of implementation.
- Providing several levels of quality of service.
- Lightweight and efficient in terms of bandwidth.
- Independence of the platform used.
- Persistence session tracking.
- Security.

The Message Queuing Telemetry Transport (MQTT) protocol, developed by IBM in 1999, fulfills all of these requirements. The web site mqtt.org provides the following protocol description: « MQTT is a simple and lightweight messaging protocol designed for limited devices and networks with low bandwidth, high latency or unreliability. It was developed on the principles of minimizing network bandwidth and device resource requirements, while trying to provide reliability and a certain degree of confidence in delivery. These principles also make this protocol ideal for the world of connected machine-to-machine or IoT devices, as well as for mobile applications where bandwidth and battery power are critical». 
As a cloud service provider, vscale.io hosting was used. On the server using the standard methods of the Linux operating system distribution, the MQTT broker software and the nginx web service were installed, and the security system was configured.

The core of the MQTT client is the Eclipse Paho JavaScript library. The Eclipse Paho project provides a number of open source clients for the MQTT messaging protocols. The Paho JavaScript client is a browser library that takes advantage of WebSockets to connect to the MQTT broker. The library was originally created by Andrew Banks at IBM and was donated by Eclipse by IBM in 2013. It should be noted that there are a sufficient number of MQTT client programs that can be installed for free in the Android operating system and make it easy to configure any displayed parameters. For quick testing of programs, a similar program MQTT Dash was used.

The network topology is shown in Figure 2. When using the MQTT broker, there can be many publishers and many consumers. The MQTT protocol successfully separates publishers from consumers. Thus, the microcontroller connected to the packaging machine is a client of the MQTT cloud service server. After collecting information from sensors and devices, the microcontroller program generates a package and publishes it in certain topics of the MQTT server, for example temperature, cycles, etc. Other clients, for example, a web browser running on a personal computer or smartphone, subscribe to these topics and accordingly display information about the current state of the machine.

The ESP32 microcontroller in the used network topology (Figure 2) transmits data using wireless technology, then, despite the reliability of TCP and MQTT, communication may be lost. In addition,
the microcontroller may lose power, lose a strong signal, or it may just be a failure, and the session will go into a half-open state. In that time as the server will assume that the connection is still reliable and expect data. The MQTT protocol reliably handles such situations using timers; the broker sends a message to all clients to close the topic. If a client tries to reopen the connection, a broker will create a new connection.

When programming a client, it is possible to set the Quality of Service (QoS) parameter. There are three options for this parameter (quality of service levels):

- QoS-0 (non-assured transmission): This is the minimal QoS level. This is analogous to a fire-and-forget model detailed in some of the wireless protocols. It is a best-effort delivery process without the receiver acknowledging a message or the sender reattempting transmission.
- QoS-1 (assured transmission): This mode will guarantee delivery of a message at least once to a receiver. It may be delivered more than once, and a receiver will send an acknowledgment back with a PUBACK response.
- QoS-2 (assured service on applications): This is the highest level of QoS that ensures and informs both the sender and receiver that a message has been transmitted correctly. This mode generates more traffic with a multi-step handshake between the sender and receiver. If a receiver gets a message set to QoS-2, it will respond with a PUBREC message to the sender. This acknowledges a message and a sender will respond with a PUBREL message. PUBREL allows a receiver to safely discard any re-transmissions of a message. The PUBREL is then acknowledged by the receiver with a PUBCOMP. Until the PUBCOMP message is sent, the receiver will cache the original message for safety [2].

Since the acceptance of a repeated message can lead to the construction of an incorrect sequence cyclediagram of the packaging machine, QoS-2 was used in client programs, both on the microcontroller and in the web application.

4. Results
The result was a fully functional remote monitoring system for the operation of the packaging machine. In the event of a breakdown of one of the sensors or a violation of the fence, the client is given an error code and its description, which allows almost instant establishment of the cause of the failure and the restoration of a machine to working condition. Thus, the integration of cloud technology in the packaging process of cooked-smoked sausages allowed reducing downtime and minimizing equipment repair costs. For example, in the welding tool of the machine there were failures of the heating elements, as a result of which the thermostat increased the operating time of the remaining heating elements and the package overheated on one side and not tight welding. Over time, the packaging lost its tightness and stores returned defective goods. The installation of a digital ammeter and data allowed instant identification and solution of this problem. Marriage declined from 8% to zero. In addition, maintenance and repair decreased from 16 hours a month to 4 hours.

The templates of certain controlled equipment parameters were created in the JavaScript programming language. The templates make it easy to create web applications for customers. The web page may display such customizable parameters as the identifier and password of the wireless network, broker's address and port, temperature correction, etc. The displayed parameters are also set. Using the developed web service will allow uniformly displaying information on various devices and in various operating systems. While the available programs of MQTT clients can only work in the Android operating system and are unsuitable for industrial use.

5. Conclusion
The observation of the developed system showed its high reliability. The selected maximum quality of service did not stop receiving data on the packaging machine in real time with minimal delay. In case of loss of wireless connection, the microcontroller was waiting for an available connection to the router and, if possible, resumed the connection, at this time the data was stored in the microcontroller buffer. When reconnecting, the microcontroller publishes all the data from the buffer. The broker
guarantees the delivery of data to signed customers. In addition, all the conditions of the packaging machine during the work shift are recorded in a text file on the server and can be read when analyzing malfunctioning or breakdowns, unplanned downtime.

The adjusters or a mechanic regularly conduct rounds to monitor the operation and timely maintenance of plant equipment. During the bypass, the health of the equipment is monitored, equipment malfunctions are detected, defects are recorded. The information collected in this way is transmitted to the repair service of a plant. In addition, it is often necessary to follow the readings of devices, for example, the readings of the ammeter, in order to determine the breakdown of the heater in the welding tool of the packaging machine. All this is done with a certain frequency during the work shift. All cases are recorded on paper.

The proposed monitoring system is more efficient and allows performing all of the above on-line and remotely.

The developed applications allow the workshop mechanic to timely see problems in the operation of equipment, adjusters to see the results of the previous shift, and fix problems. The application allows the chief mechanic to solve the problem of collecting a large amount of information about the work of all the shops of the enterprise for further analysis and management decisions based on it.

The system is well protected. For communication on the Internet, applications use the Secure Sockets Layer (SSL) cryptographic protocol [8]. In the future, it is planned to use the Virtual Private Network (VPN) technology. There is also a password protection system at the application level and access for users with different access levels, for example, an installer can only observe temperature readings, and the mechanic can make a temperature correction, only the chief mechanic has access to the equipment operation log.

Reference
[1] Shishkina D I and Sokolov A Y 2018 Analysis of foreign technologies for the functional meat products Proceedings of the Voronezh State University of Engineering Technologies 80(2) 189-194
[2] Perry Lea 2018 Internet of Things for Architects (Birmingham: Packt Publishing)
[3] Matthew J Perry 2016 Evaluating and Choosing an IoT Platform (Sebastopol, California: O'Reilly Media)
[4] Ryan Betts 2016 Architecting for the Internet of Things (Sebastopol, California: O'Reilly Media)
[5] Syed Zaeem Hosain 2018 The Definitive Guide. The Internet of Things for Business (San Jose, California: Aeris Communications)
[6] Rusanov V V, Perov V I and Samoilov M A 2018 Automation of public catering enterprises using modern digital technologies: Arduino IDE, OPC Modbus and Master Scada programs Proceedings of the Voronezh State University of Engineering Technologies 80(2) 38-44
[7] Agus Kurniawanb 2019 Internet of Things Projects with ESP32 (Birmingham: Packt Publishing)
[8] Fei Hu 2016 Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations (Boca Raton: CRC Press)