EMBEDDED IOT PLATFORM FOR REMOTE TRAFFIC CONTROL IN SMART CITY IOT INFRASTRUCTURE

Serhiy Nikolskiy, PhD Student, Iryna Klymenko, Dr.Sc., As.-Prof.
The National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine;
e-mail: serhiy.nikolskiy@gmail.com

Abstract. A justified hybrid multilevel approach to IoT infrastructure implementation is set to facilitate the achievement of a scalable IoT (Internet of Things) infrastructure, incorporating integration into cloud technologies and services. The localization of hardware-software traffic management means at the lower level of the IoT infrastructure, close to data collection devices, ensures the generation of control influences in real-time, and relieves communication channels at the higher levels of the IoT infrastructure architecture. A technology for generating control influences for remote traffic management is proposed, which is based on the developed AT command system for deploying a web server and generating web pages using the capabilities of the embedded IoT platform on modern microcontrollers. The proposed technology allows for the formation of control influences in real-time, in an easily comprehensible textual format, using a web interface in the local domain of the IoT infrastructure. It also enables the visualization of information on remote displays and information boards, as well as on displays integrated into automotive equipment. The proposed technology can be used to inform road traffic participants about critical situations and can be embedded in smart traffic lights within remote traffic management systems or used to implement virtual traffic lights.

Key words: Embedded systems, Internet of Things, Edge Computing, Microcontroller Units, remote control, traffic control, Smart City

1. Introduction

Advancements in wireless communication channels in recent years have opened up prospects for utilizing the Internet to popularize services. Cloud technologies and services have significantly influenced this progress, offering the potential for unlimited scalability of computing power, data storage capacity, analytic services, and monitoring. To meet ever-growing human needs, the issue of developing information infrastructure is becoming increasingly significant. This arises from the necessity of establishing communication among a large number of devices, which often surpasses the number of users in modern infrastructures. Presently, the most popular solution to this problem is the IoT. The IoT paradigm, actively implemented and becoming an integral part of the human environment, is primarily aimed at executing numerous processes without human involvement.

This justifies the current issue of interaction between things without human intervention in the IoT paradigm, which involves the automated collection of data from distributed and remote sensors, processing and analysis of vast volumes of heterogeneous data, including intelligent analysis and real-time support [1]. Process automation in the IoT paradigm makes it possible to develop probabilistic behavioral scenarios and environments that can enhance the comfort and safety levels in human life, ultimately significantly influencing the development of human society as a whole. The realization of the Smart City concept is closely associated with the IoT paradigm today [2, 3]. This concept forms the basis for the prospective development and optimization of electronic governance systems and the automation of day-to-day management in cities all around the world.

As urbanization processes increase, the number of vehicles on the roads also rises, leading to traffic congestion and the emergence of traffic jams. One of the current directions of IoT infrastructure development relates to traffic management. Various automated remote traffic control systems have been described in literature sources [2, 4, 5], including remote control based on "smart traffic lights" [6], which integrate into the IoT infrastructure of Smart Cities. Advanced challenges in this field involve replacing physical traffic lights with virtual ones. Under such conditions, reducing human intervention would contribute to reducing the number of errors caused by the "human factor" during dispatching traffic control. The intelligence in such systems is based on the development of algorithms and methodologies for the functioning of automatic data collection and processing means in the IoT infrastructure [3].

Automated remote control in the IoT paradigm is based on the use of wireless technologies and small to medium-range communication channels [7]. Wi-Fi wireless technologies today handle a significant portion of wireless traffic from various IoT devices [8, 9], driving the rapid development of Wi-Fi to ensure exceptionally high data throughput and reliability in IoT infrastructure [3].

2. Problem statement and Drawbacks

The theme of remote control is relevant within the paradigm of modern smart environments based on IoT technologies. This theme has been extensively explored in various literature sources. In the work [10], the characteristic features of implementing remote control systems are generalized, particularly focusing on their hardware components.

To implement automated remote control means within the IoT, a variety of wireless communication technologies in the Internet network are utilized. The choice of data transmission technologies by sensors directly influences the productivity of IoT applications. Diverse information collection device technologies are applied within the Internet of Things (IoT) infrastructure. This particularly applies to IoT devices that are integral parts of more complex solutions, such as smart cities or intelligent control systems. Many sensors specifically designed for narrow application ranges are widely used to address relevant tasks in IoT [11]. In the work [7], a detailed overview and analysis of short- and medium-range wireless communication technologies for IoT applications are provided. It's noted that the most common wireless interfaces for implementing remote communication in complex infrastructures and control systems are radio channels of varying ranges and frequency bands, such as ultra-wideband (UWB), Radio Frequency Identification (RFID), Bluetooth, Wi-Fi, and ZigBee. Bluetooth is used for short-range wireless data transmission between IoT devices. For medium-range communication, technologies like 2G, 3G/GSM, 4G/LTE, and 5G are preferred. Wi-Fi technology occupies a leading position, primarily due to its low latency, high data transmission speed, and extended range. In the work [3], the authors mention that future IoT devices will increasingly depend on subsequent generations of Wi-Fi that will be more powerful, efficient, and sophisticated. Presently, Wi-Fi technologies handle a significant portion of wireless data traffic from various devices, and the sixth generation of Wi-Fi (Wi-Fi 6) signifies enhanced communication efficiency while utilizing the recently available 6 GHz spectrum (Wi-Fi 6E). Efforts towards standardizing the seventh generation (Wi-Fi 7) aim to provide extremely high throughput (IEEE 802.11be) [8, 9].

However, within the IoT infrastructure, data transmission channels' implementation serves as an extensive means of enhancing efficiency. Currently, IoT infrastructure developers are considering a complex set of measures to improve IoT architecture and develop efficient software and hardware solutions. The primary conceptual approach that is well-founded in the IoT paradigm is the placement of part of the infrastructure at the edge of the IoT network (EC, Edge Computing). Given the relevance of cloud technologies, which provide powerful data processing and analytics services, storage, and scalable distributed computational power, a hybrid approach to IoT architecture is gaining prominence. This approach integrates both cloud and edge technologies. The hybrid architecture, which represents the synergy of cloud and local IoT environments, is described in the work [12]. Architectural enhancements using edge computing transfer a portion of computations and data storage to the data source, contributing to improved data collection and analysis efficiency by reducing response time, ensuring real-time operation, preserving bandwidth, increasing result reliability, enhancing security, and reducing energy consumption. Solutions based on the hybrid approach and Edge Computing are extensively covered in the literature dedicated to the development of smart spaces. Smart parking lots [1, 4], traffic management systems [12], and intelligent traffic lights [6] are the most characteristic examples. The challenges of enhancing the efficiency of real-time data collection and analysis processes, improving hardware and software means for data collection, analysis, and control influence generation at the IoT boundary, as well as the development of edge computing services, are relevant tasks addressed in this article.

3. Goal

The objective of the research is to develop conceptual and hardware-software means for implementing remote traffic control using wireless data transmission channels, specifically Wi-Fi. The research aims to create hardware-software solutions at the local domain level of the IoT infrastructure, which can be integrated into the global existing smart city infrastructure built within the framework of the hybrid IoT architecture concept, based on edge computing technology.

4. Materials and methods of the remote-control means development

The tasks set are as follows:

− Justify the concept of implementing a remote traffic control system with the potential for integration into the global IoT infrastructure of a smart city for further implementation.

− To address the remote-control task, develop hardware and software means for implementing an embedded IoT platform capable of facilitating automated data collection and the transmission of control inputs within the local domain of the IoT infrastructure.

− Develop an experimental setup for the implementation and configuration of the embedded IoT platform.

4.1. Justification of the remote traffic control concept in a Smart City for the hardware-software means implementation

The approach of a smart traffic light utilizing advanced technologies and specialized algorithms that consider the flow of cars and bicycles based on the intersection's conditions is discussed in [6]. The system takes into account the road load and switches the traffic light signal depending on the number of vehicles. The switching parameters of the traffic light are the road load and the overall carbon dioxide emissions, which are sensed by sensors and calculated by the system. The innovative idea of replacing physical traffic lights with virtual ones is also presented. To implement this concept, the authors propose using a microcontroller that functions as a smart traffic light. The smart traffic light, in turn, becomes a part of the entire IoT infrastructure, including integration with cloud technologies. The smart traffic light serves as a microcomputer with a transceiver connected to the global infrastructure. On the other hand, smart traffic lights act as relays in the wireless global network, communicating with onboard vehicle computers. While the authors do not delve into hybrid architectures, the proposed cutting-edge solution serves as the foundation for the research conducted in this article.

A well-known method of intelligent traffic management, which could potentially integrate the developed remote control means, is described in [12]. The idea is based on deploying sensor networks at each road intersection. Hybrid sensors can be used for monitoring the number of vehicles by detecting their presence at specific locations and identifying vehicles by their registration numbers using RFID wireless communication tags. Presence sensors placed on traffic lanes at each intersection are employed to determine traffic density and control traffic lights to enhance traffic flow at that intersection. The method relies on deploying a cluster network topology, allowing various types of sensors placed at each intersection to transmit their data to the global control center of the system through a relevant gateway.

All of this fits into the theory of hybrid organization of IoT architecture. We consider the local domain at the level of each network cluster. The edge router serves as a gateway for accessing the global control center. The embedded system based on a microcontroller performs tasks of collecting and locally analyzing information from sensors and sensors within the local domain - the cluster. Communication with cloud services occurs through the control center. The control center also can connect with the proposed technical solutions in each local domain for transmitting control influences and messages, although this goes beyond the scope of the research in this article.

Figure 1 illustrates the concept of remote traffic management in a smart location for further implementation. The concept is developed based on the analysis of known systems for remote automated traffic management and the implementation of smart traffic lights. The hybrid three-tier architecture, which is considered the most relevant, is depicted in Figure 1. The first level represents the local domain where the sensor and sensor system are localized. The local control center, which functions as a data aggregator, local analytics server, and smart traffic light control device, is connected to all devices within the local domain and the upper levels of the IoT infrastructure. The central control center serves the function of integrating local domains and can be implemented either at the boundary of the second level of the IoT infrastructure or at the level of cloud technologies and services.

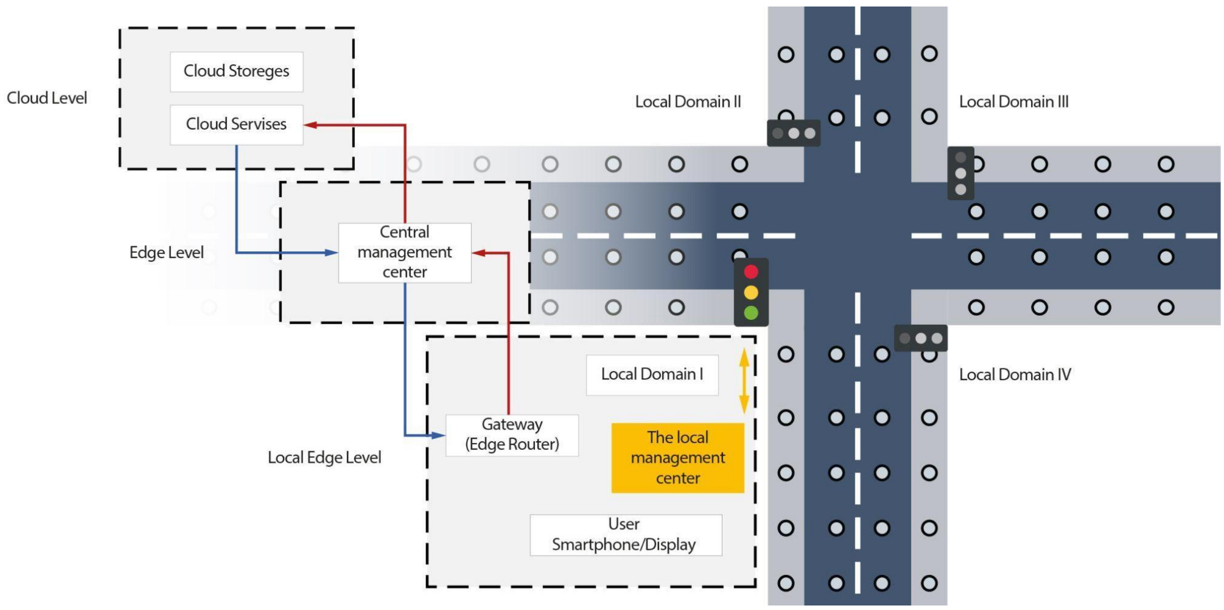

Fig. 1 Structural diagram of the implementation of a three-level hybrid IoT architecture for remote traffic management.

4.2. Hardware and software means for the embedded IoT platform development

For the development of the experimental prototype of the embedded IoT platform, the Global Logic StarterKit development board [13] was used. The development board includes an MCU (Microcontroller Unit) STM32F407 Discovery with the STM32F407VGT6 microcontroller from STMicroelectronics (a European international company based in France and Italy). It also features an expansion board containing an embedded graphical display and a set of necessary sensors. During the experiments, temperature sensors, air quality sensors, and an accelerometer were utilized.

The Global Logic StarterKit development board does not include the software and hardware necessary for connecting to the IoT infrastructure. For IoT connectivity, one of the most popular MCUs for IoT applications was employed – the ESP8266 with a Wi-Fi module. To connect to the IoT, one of the most widely used standalone Wi-Fi transceivers, the ESP8266 controller, was utilized. For collecting sensor data and connecting to the IoT infrastructure, wireless data transmission technology Wi-Fi [7] was used, which allows for achieving the required signal range within the local IoT domain and provides a satisfactory level of energy consumption for the developed IoT solution.

The ESP8266 MCU, in its default configuration, is loaded in serial modem mode. In this mode, communication with the ESP8266 module is possible using a set of AT commands. The AT commands for ESP8266 are divided into Wi-Fi layer commands and TCP/IP layer commands. The set of AT commands serves as the effective communication standard with the ESP8266 module. The use of AT commands to establish a connection with the ESP8266 module is also detailed by the authors in [4]. Essentially, AT commands provide an alternative to a command-line interface for historical routers. However, these solutions remain relevant today and are often used in devices utilizing GSM, Wi-Fi, and Bluetooth for configuration and communication with the host.

5. Development of the embedded IoT platform for data collection from sensors

The structure of the embedded IoT platform is presented in Figure 2. The architecture of the local domain of the IoT infrastructure consists of an embedded IoT platform based on the MCU STM32F407 Discovery, which is connected to the ESP8266 module through the UART interface.

For remote interactive connection with the ESP8266 module, a technology utilizing AT commands is proposed. The MCU functions as a host, sending AT commands and waiting for responses or other notifications from the ESP8266 module. Through the ESP8266, the MCU automatically connects to the operational network using the specified Service Set Identifier (SSID) of the wireless network, becoming a web server. Other hosts within the local IoT network infrastructure connect to this web server and receive a statically created web page with real-time sensor readings. Nodes within the local network can include users with mobile devices running an application, in-car computers, other embedded systems within vehicles, boundary servers for data processing and storage, local dispatch stations, and more. The structural integration of the experimental setup into the infrastructure of the local IoT domain is illustrated in Figure 2.

The structural diagram of the proposed software solution and the interconnection of key software modules are shown in Figure 3. The software for the embedded IoT platform based on the STM32F407 Discovery MCU is implemented on the FreeRTOS operating system. For the development or integration of drivers from other sources, the STM32 HAL embedded software was utilized. The STM32F407 controller was configured using the STM32 CubeMX development environment. STM32CubeMX is a graphical interface tool provided by STMicroelectronics specifically designed for the automatic configuration of STM32 devices and for generating C language source code.

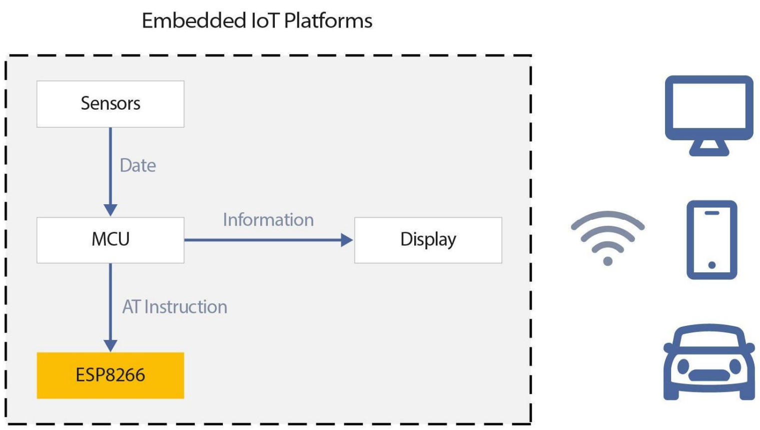

Fig. 2 Structure of the embedded IoT platform

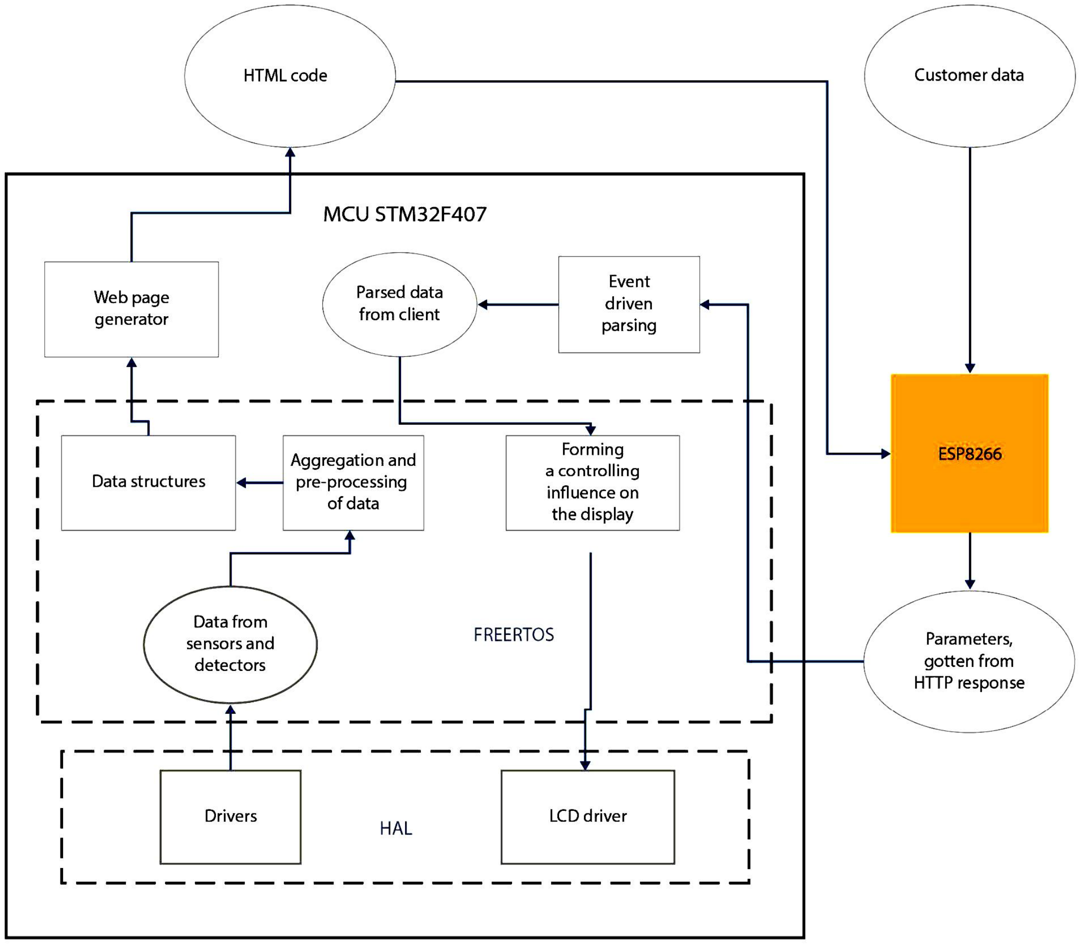

Fig. 3 Structural diagram of the software solution

The proposed technology for remote interactive connection to the ESP8266 module is implemented through a system of AT commands. To develop the command system for creating and operating the web server, as well as for debugging and experimentation, the remote terminal tool "picocom" was used. The first step is to establish a connection with the ESP8266 module using AT commands. When a command is entered in the terminal, information about the sent message (echo) is displayed, and upon pressing ENTER, the response "OK" is received. The developed command system is based on the utilization of a standardized set of commands. For experiments involving connection and performance assessment of the system in a network environment, the command system incorporates basic command types with necessary parameters:

AT+<x>=? – test

AT+<x>? – query

AT+<x>=<...> – configuration

AT+<x> – command without parameters

Executing the provided commands helps to understand that the solution is functional. One notable advantage of AT commands is their linguistic interface, which is easily understood by humans and enables the creation of parsers at the user-level software.

Commands are created using the C language, as shown in the previous experiment, and sending a command in a remote terminal is straightforward. The challenge arises when receiving the execution results of a command. This is why a command system and an experimental methodology using AT commands to control the ESP8266 module were proposed.

Experiment 1. Let's consider the need to connect to an access point with access parameters ssid='Hello' and psk='World'. The goal is to determine the state of the module, whether it is operational, and whether the access point for connection is available.

The connection methodology is divided into the following stages:

1. **AT:** Checking the response to the AT command.
2. **AT+RST:** Resetting the module to its initial state. (Waiting for the "ready" response)
3. **ATE0:** Turning off echo (Waiting for the "OK" response)
4. **AT+CWMODE=1:** Setting the mode to station - client. (Waiting for the "OK" response)

5. **AT+CWJAP="Hello", "World":** Connecting to the access point. (Waiting for the "WIFI CONNECTED" response)

Experiment 2. In the case where the ESP8266 module is already connected to a Wi-Fi access point. The command **AT+CWJAP?** allows you to check which Wi-Fi network the module is currently connected to and whether it's connected at all. The response will be in the format **+CWJAP:ssid**. However, in this case, it's better not to send the **AT+RST** command.

Experiment 3. Methodology for creating a web server based on the ESP8266 module.

AT+CIPMUX=mode - Connection to multiple (up to 4) connections for the server. The command to set the multiple connections mode allows enabling the server:

AT+CIPMUX=1 sets the multiple connections mode.

AT+CIPSERVER=mode, port sets the server mode. After this, other devices that know the module's address can establish connections with it. The command takes the following values as parameters: port - 80, create server - 1. Therefore, the following command allows setting the server mode:

AT+CIPSERVER=1,80.

Experiment 4. After creating the web server, we await messages. To read messages, it's necessary to set up interrupts on the UART port. This was done as follows: when a new 8-bit number arrives on the line, an interrupt is executed, and the number is written to a circular buffer. The thread reads a string from the buffer and looks for indicators that allow determining the types of incoming data. The data coming from the module looks like this:

WIFI <indicator word> ready - the module is ready to work.

busy p. <id> - not ready.

CONNECT - client connected indicator <id>.

CLOSED - client disconnected indicator.

To implement the project, a modified firmware for ESP8266 was developed. STM32CubeMX development environment was used for compiling and flashing the embedded system. After configuring the firmware settings, it's flashed onto STM32F407 Discovery, and it starts working and looking for a connection with the ESP8266 module.

6. Discussion of Research Results

The developed software provides the capability to create a web interface for sensors, sensors, and displays in the IoT infrastructure. The developed software solution establishes a web server at the embedded system level within the local IoT infrastructure. The connection scheme between the embedded Global Logic StarterKit system and ESP8266 can be seen in Figure 3, and experiments with hardware equipment are depicted in the photo in Figure 4. In [13], the authors investigate the effectiveness of modifying the proposed solution using a web interface to implement remote monitors and bulletin boards. The described solutions can be utilized to create means for transmitting control inputs in textual form to monitors or displays of automotive computers. An example of generating a web page on a remote IoT device (such as a dispatcher's console) is shown in Figure 5. The web page is configured to refresh automatically every 4 seconds or manually as needed.

The experimental prototype's web page displays sensor readings in real-time, including temperature, air pollution levels, and accelerometer axis measurements. A sophisticated feature for traffic management involves creating a text string that can be input by an operator through a form on the web page. Additionally, textual information for conveying control actions to clients can be generated automatically based on processed data. Using this text string, notifications can be sent about traffic jams or congestion in specific geographical areas within each local IoT infrastructure domain.

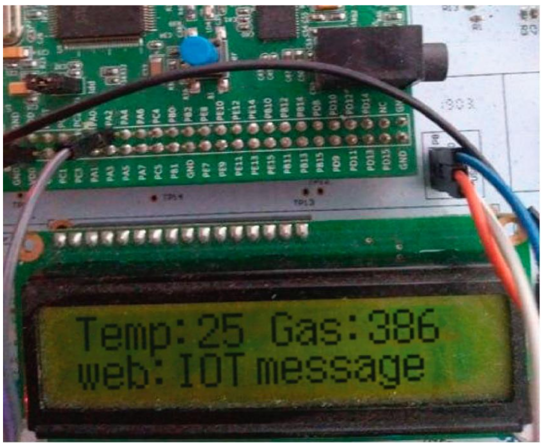

Fig. 4 Display Connection for Transmitting Control Inputs

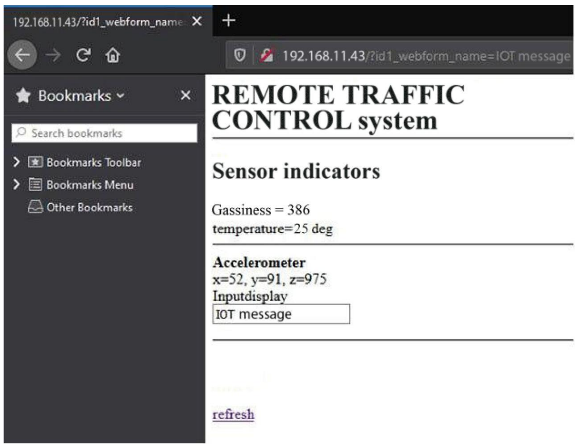

Fig 5 Example of Web Page Generation

A proposed software solution has been introduced to implement the functional capability of data collection and the formation of control influences in the form of textual information. The developed means are realized at the level of the local domain within the IoT infrastructure, as illustrated in Figure 1. At this local domain level, processes for collecting data from sensors and sensors are taking place. The specific processes of gathering data from sensors were not addressed in this article, but they could be subjects for further research. The future roadmap envisions the implementation of a hybrid multi-tier IoT architecture, leveraging cloud services for further data analysis and the integration of local domains into the global IoT infrastructure.

The developed experimental prototype utilizes a limited number of sensors. The concept of a local edge web server remains relevant in the context of advancing technologies in edge computing, reliability, real-time data processing, and cost-effectiveness. Drivers collect data and organize it into a predefined structure, which also includes a finite state machine for the web server. Subsequently, the data is copied onto the generated web page, and stored in the RAM as text. Another task configures the ESP8266 for server initialization and Internet connection. If the ESP8266 successfully establishes a connection, the task responsible for handling HTTP and other requests from the ESP8266 can receive connections from other devices. Once this connection is established, the web page is updated with new values and transmitted to the client.

The developed hardware and software solution can be easily transferred to any hardware device using more powerful mini-computers designed for universal operating systems. This extension can include the addition of more sensors and detectors to address tasks related to remote control of road vehicles. By employing robust controllers, intelligent analytics can be implemented at the local domain level, enhancing the system's responsiveness in real-time scenarios. Cloud-based services would be suitable for statistical analysis and prediction of critical situations. Additionally, local devices can be entrusted with the task of adjusting statistical data and providing emergency responses.

7. Conclusions

1. By localizing the traffic control mechanisms at the lower levels of the IoT infrastructure, close to data collection devices, and utilizing Wi-Fi wireless data transmission technology within the local IoT domain, the system achieves real-time data processing from IoT sensors and relieves communication channels in the global network at higher levels of the IoT architecture. This multilevel approach enables the development of scalable IoT infrastructures for traffic management in a smart city system, with future integration into cloud technologies and services.

The developed hardware and software means for remote traffic control contribute to the enhancement of existing traffic management systems by enabling automated data collection and aggregation from sensors at the edge of the local IoT infrastructure.

2. To boost remote traffic control systems, a novel technology is proposed for generating control measures. In contrast to traditional approaches, this technology leverages the developed AT command system to deploy a web server and create web pages using the capabilities of the embedded IoT platform based on microcontrollers. This innovative approach facilitates the real-time generation of control measures in a user-friendly text format through a web interface within the local IoT domain infrastructure. It enables visual representation of information on remote displays, informational boards, and built-in vehicle displays.

3. The proposed solution provides a remote web interface and the capability for remote control of sensors with information visualization via the web interface for any IoT device within the local network.

8. Gratitude

The authors express their gratitude to the Staff members of the Department of Information-Measuring Technologies of Lviv Polytechnic National University.

9. Mutual claims of authors

The authors declare the absence of any financial or other potential conflict related to this work.

References

[1] I. Klymenko, A. Gaidai, S. Nikolskyi, V. Tkachenko, “The Architectural Concept Of The Monitoring System On The Basis On A Neuron Module IoT Data Analytics”, Adaptive systems of automatic control, p. 111-123, 2022, doi:10.20535/1560-8956.41.2022.271355.

[2] Lim C., Kim K.-J., Maglio P., “Smart cities with big data: Reference models, challenges, and considerations”, Cities, p. 86–99, 2018, doi:10.1016/j.cities.2018.04.011.

[3] A. Bagwari, J. Bagwari, T. Anand, B. Chaurasia, R.P.S. Gangwar, M. Kamrul Hasan, “The Role of IoT in Smart Technologies”, 2022, doi:10.1201/9781003045809-6.

[4] F. Al-Turjman, A. Malekloo, “Smart parking in IoT-enabled cities: A survey”, Sustainable Cities and Society, 2019, doi:10.1016/j.scs.2019.101608.

[5] A. Hilmani, A. Maizate, L. Hassouni “Automated Real-Time Intelligent Traffic Control System for Smart Cities Using Wireless Sensor Networks”, Wireless Communications and Mobile Computing, 2020, doi:10.1155/2020/8841893.

[6] V. Miz, V. Hahanov, “Smart traffic light in terms of the Cognitive road traffic management system (CTMS) based on the Internet of Things”, Proceedings of IEEE East-

West Design & Test Symposium (EWDTS 2014), 2014, doi:10.1109/EWDTS.2014.7027102.

[7] G. Oguntala a, R. Abd-Alhameed a, S. Jones a, J. Noras a, M. Patwary b, J. Rodriguez, "Indoor location identification technologies for real-time IoT-based applications: An inclusive survey", Computer Science Review, 2018, p. 55-79, doi:10.1016/j.cosrev.2018.09.001.

[8] M. Alias, D. Dzaki, N. Din, S. Deros, R. Passarella, A. Chaai, "IoT-Based Transmission Tower Monitoring Communications and Visualization Platform", 2022 IEEE Symposium on Future Telecommunication Technologies (SOFTT), 2022, doi:10.1109/SOFTT56880.2022.10010 295.

[9] A. Ali, M. Khan, N. Bolong, K. Maarof, S. Tanalol, "Conceptual and design framework for smart stormwater filtration", 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), 2020, doi:10.1109/IICAIET49801.2020.9257828.

[10] Z. Tengfei, L. Qinxiao, M. Fumin, "Remote control system of smart appliances based on wireless sensor network", 25th Chinese Control and Decision Conference (CCDC), p. 3704–3709, 2013, doi:10.1109/ccdc.2013.6561592.

[11] L. Vukonić, M. Tomić, "Ultrasonic Sensors in IoT Applications", 45th Jubilee International Convention on Information, p. 415-420, 2022, doi:10.23919/MIPRO55190.2022.

[12] A. Hilmani, A. Maizate, L. Hassouni, "Automated Real-Time Intelligent Traffic Control System for Smart Cities Using Wireless Sensor Networks", Wireless Communications and Mobile Computing, 2020, doi:10.1155/2020/8841893.

[13] I. Muraviov, V. Taraniuk, I. Klymenko, "Making an IoT development platform from a simple microcontroller demonstration board", Information, Computing, and Intelligent systems, p. 68 – 76, 2020, doi:10.20535/2708-4930.1.2020.216065.