

ДОДАТОК А

Апробація наукових результатів досліджень



Co-funded by the
European Union

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки

Харківський національний університет міського господарства

імені О.М. Бекетова

Братиславський університет економіки та менеджменту

Громадська організація «Перспектива»

Угорський університет сільського господарства та природничих наук

МАТЕРІАЛИ

I International Conference

**«Sustainable smart cities and communities:
business and innovation solutions»**

(Сталі розумні міста та спільноти:

бізнес та інноваційні рішення)

SSC&C2025

21 квітня 2025

[електронне видання]

Харків 2025

ЗМІСТ

<i>Vladyslav Yevsieiev</i>	
Mobile robots and autonomous vehicles in the mobility as a service (MAAS) concept	7
<i>Svitlana Starykova</i>	
Automation of urban infrastructure based on predictive maintenance and IoT	9
<i>Oleksii Fomin</i>	
Development of an automated system for the technological processes of a "concrete plant" (a company for the production of construction components)	11
<i>Nazarii Piven</i>	
Free software as a tool for technological advantage and independence in the digital environment	13
<i>Igor Golod</i>	
Інтелектуальні системи підтримки прийняття рішень для оптимізації мікроклімату в промислових умовах	15
<i>Dmytro Gurin</i>	
Intelligent tracking algorithms in collaborative robotic systems: application of camshift and kalman filter	17
<i>О.О. Гуртовий</i>	
Аналіз сучасного стану обліку розподілених витрат тепла	20
<i>Valeriia Darahan, Irina Kolupaieva, Yurii Romashov</i>	
A general approach to develop digital twins of automation objects for smart cities applications	22
<i>Д. С. Заяць</i>	
Розпізнавання жестів і комп'ютерний зір для безконтактного керування пристроями ...	24
<i>Illia Kalashnykov, Iryna Kolupaieva, Yurii Romashov</i>	
Resistance sensors of angular velocity for research robots to benchmark smart cities applications	26
<i>Irina Kolupaieva, Yurii Romashov, László Vértesy</i>	
Implementations of circular economy principles for sustainability of laboratories in universities inside smart cities	28
<i>Iryia Kolupaieva, Igor Nevliudov, Yurii Romashov</i>	
European views on a green automation as a crucial technology for circular economy implementations to smart cities	30
<i>Irina Kolupaieva, Igor Nevliudov, Yurii Romashov</i>	
European views on an intelligent automation as a crucial technology for circular economy business models in smart cities	32
<i>Irina Kolupaieva, Yurii Romashov</i>	
High-performance computing to research resource and energy efficiencies of automated controls for smart cities applications	34
<i>С. В. Хрустальова, Н. Р. Курбанов</i>	
Перспективи розвитку альтернативних систем електрогенерації	36
<i>R.V. Marunich, S.V. Sotnik</i>	
Modern IoT technologies for creating automated access systems	38
<i>В.О. Михайлов, І.В. Білецький</i>	
Впровадження інноваційних цифрових рішень у інфраструктуру закладів охорони здоров'я	40

DEVELOPMENT OF AN AUTOMATED SYSTEM FOR THE TECHNOLOGICAL PROCESSES OF A "CONCRETE PLANT" (A COMPANY FOR THE PRODUCTION OF CONSTRUCTION COMPONENTS).

Oleksii Fomin

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky av., 14

E-mail: oleksii.fomin@nure.ua

Annotation: This work presents the development of an automated system for a concrete plant, which includes the development of control system software to optimize the processes of mixing, transportation, and storage of concrete mixtures. The effectiveness of the proposed system has been studied and analyzed in terms of quality control of technological processes, reporting on the operations of concrete mixers, and calculating the materials used. The results obtained can be useful for engineers and developers in the field of production process automation.

Key words: Automated system, software, concrete plant server, control system, process optimization, quality control, reporting, concrete mixer, production materials, automation.

In the current conditions of industrial development, automation of technological processes has become one of the key factors in improving production efficiency and reducing costs. Concrete plants, which specialize in the production of various types of concrete mixtures, constantly face numerous challenges related to optimizing production processes, ensuring high product quality, and reducing resource consumption. This includes issues related to equipment management, material and product quality control, as well as the efficient use of energy and labor resources. Automation of technological processes at a concrete plant is an important step not only for increasing overall production productivity but also for achieving significant reductions in energy costs and labor intensity, which in turn helps to substantially lower the cost of the finished product. Furthermore, an automated system allows for detailed control over material usage throughout the entire production cycle—from the manufacturing stage to the storage and delivery of finished mixtures to end consumers.

The project was implemented using a number of advanced technologies, including PHP, React.js, Python, and MySQL. PHP handles the server-side part of the system, request processing, and interaction with the database. React.js is used to create an interactive user interface that ensures fast and convenient interaction with the web application. Python is responsible for acting as an intermediary between the server and the control panel, forming statuses, ttms, and so on. MySQL is used to store and manipulate data, providing a reliable and scalable database for storing all necessary information. The MQTT protocol is used to ensure secure and fast communication between systems, especially for interacting with control panels and other devices in real time.

An important element of the project is the use of advanced visualization technologies that allow the creation of dynamic and interactive graphical interfaces. These technologies make it possible to effectively display data, providing users with clear and understandable representations of information in the form of graphs, charts, and other visual information, which helps in decision-making and process optimization.

The development of an automated system for the concrete plant will include the creation of software for quality control, raw material usage, and reporting. By implementing such innovative approaches, production will not only be faster and more productive but also capable of providing accurate reports on the materials used, the status of the automated mixers, and the quality level of the product at each stage of production. The main goal of this qualification work is to develop a system that ensures effective control over the quality of technological processes at the plant, automates the storage of data on the materials used, and allows for the rapid generation of reports on the operation of technological equipment, significantly simplifying managerial tasks.

Additionally, the implementation of this automated system will enable the integration of real-time data analysis and predictive maintenance. By constantly monitoring the performance of equipment and material consumption, the system will be able to predict potential malfunctions or inefficiencies before they occur, allowing for proactive interventions. This predictive approach will not only prevent downtime but also extend the lifespan of the plant's equipment. Moreover, the integration of data analytics will provide valuable insights into production trends, enabling management to make informed decisions that further optimize processes and reduce costs. This holistic approach to automation will ensure that the plant remains competitive and capable of meeting increasing demands while maintaining high operational standards.

Thus, the development of an automated system for technological processes at a concrete plant is an extremely important and relevant task that contributes to improving production efficiency, enhancing product quality, and significantly reducing costs at all stages of the production cycle. The implementation of such technologies will have a major impact on optimizing all processes, making production more flexible and stable. This will be an important step towards modernizing concrete plants and significantly improving their competitiveness in the market, allowing companies to adapt to changes in the external environment and market demands, increasing their resilience to economic fluctuations and technological challenges.

CONCLUSIONS. Automation of concrete plant processes increases efficiency, reduces costs, and ensures quality control. Implementing a management system optimizes production and material accounting, contributing to cost reduction and productivity growth.

References:

1. "Automation and Computer-Integrated Technologies" of the Educational Program "Automation and Computer-Integrated Technologies": Textbook / I. Sh. Nevlyudov, O. I. Filipenko, O. V. Tokarieva, S. P. Novoselov, O. V. Sychova. Kharkiv: Ivanchenko I. S. Publishing House, 2023. – 151 pages.
2. Nevliudov, I., Yevsieiev, V., Maksymova, S., Demska, N., Kolesnyk, K., & Miliutina, O. (2022, September). Object Recognition for a Humanoid Robot Based on a Microcontroller. In 2022 IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH) PP. 61-64. DOI: 10.1109/MEMSTECH55132.2022.10002906.
3. Methodical Guidelines for Practical Classes and Independent Work in the Discipline "Business Process Automation" for Students of All Forms of Study in the Specialty 151 – Automation and Computer-Integrated Technologies Educational Program / compiled by: Sh. A. Omarov, N. V. Kosenko, A. I. Kolomyets; Ministry of Education and Science of Ukraine, KhNURE. – Kharkiv: KhNURE, 2020. – 24 p. – 1.70.
4. Attar, H., Abu-Jassar, A. T., Yevsieiev, V., Lyashenko, V., Nevliudov, I., & Luhach, A. K. (2022). Zoomorphic mobile robot development for vertical movement based on the geometrical family caterpillar. *Computational intelligence and neuroscience*, 2022(1), 3046116.
5. Kateryna Nimets¹, Amer Abu-Jassar², Svitlana Maksymova¹, & Vladyslav Yevsieiev, . (2025). A SMALL-SIZED MOBILE ROBOT DEVELOPMENT . *ACUMEN: International Journal of Multidisciplinary Research*, 2(3), 12–25. Retrieved from <https://inlibrary.uz/index.php/aijmr/article/view/72290>
6. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2025). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. *International Journal of Crowd Science*, 9(1), 29-43.
7. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Demska, N. (2025). Development of a model for recognizing various objects and tools in a collaborative robot workspace. *ACUMEN: International journal of multidisciplinary research*, 2(1), 224-239.

ДОДАТОК Б

Програмна реалізація підключення та зберігання даних з топіка через MQTT протокол

```
from http import client
import json
from django.utils import timezone
from .models import Ttnstate, Report, Product, Reportcurrentloop,
Ttn
import uuid
import paho.mqtt.client as mqtt
from django.db import connection
import datetime
import sys
import os
import io
from django.db.models import Q

sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')
client = mqtt.Client()

# Налаштування MQTT
MQTT_BROKER = "191.101.2.61"
MQTT_PORT = 1883
MQTT_USERNAME = 'szm'
MQTT_PASSWORD = 'sdda8jKHks3'
MQTT_CLIENT_ID = str(uuid.uuid4())
MQTT_TOPICS_SUB = [
    ("grand_beton/report/1", 0),
    ...
    ("grand_beton/production/1024", 0)
]

MQTT_TOPIC_PUB = "grand_beton/ttn/back"
MQTT_QOS = 0

def ensure_connection():
    if connection.connection and not connection.is_usable():
        connection.close()

def save_message(client, topic, payload):
    ensure_connection()
    data = json.loads(payload)

    if "ttn" in topic or "production" in topic:
        code = data.get("st", None)
        date = data.get("datetime", None)
```

```

# Check if date is a string before trying to convert it
if isinstance(date, str):
    date =
timezone.make_aware(timezone.datetime.strptime(date, "%Y-%m-%d
%H:%M:%S"), timezone.get_current_timezone())

# Remove the timezone for saving
date = date.replace(tzinfo=None)

ttn_list = data.get("ttn", [])

if isinstance(date, datetime.datetime):
    naive_datetime = date
else:
    naive_datetime = datetime.datetime.strptime(date, "%Y-
%m-%d %H:%M:%S")

aware_datetime = timezone.make_aware(naive_datetime,
timezone.get_current_timezone())

for ttn in ttn_list:
    idTtn = ttn.get("ind_ttn", None)
    idProduct = ttn.get("ind_product", None)

    try:
        if idProduct is not None:
            idProduct = int(idProduct)

        ensure_connection()

        existing_record = Ttnstate.objects.filter(
            Q(state=code) & Q(idTtn=idTtn)
        ).exists()

        if not existing_record:
            Ttnstate.objects.create(idTtn=idTtn,
state=code, date=date, json=payload)
            print(f"Saved new Ttnstate: idTtn={idTtn},
state={code}, date={date}")
        else:
            print(f"Record with ttnState idTtn={idTtn},
state={code}, date={date} already exists")

        ensure_connection()

        if Ttn.objects.filter(id=idTtn).exists():
            update_data = {"state": code}
            if idProduct is not None:

```

```

update_data["idProduct"] = idProduct

Ttn.objects.filter(id=idTtn).update(**update_data)
    print(f"Updated ttn: {update_data}")
else:
    print(f"No record found in ttn for
idTtn={idTtn}")

except Exception as e:
    print(f"Error: {e}")

if code == 14:
    bsu = data.get("bsu", None)
    new_payload = json.loads(payload)
    new_payload['st'] = 15
    new_topic = f"grand_beton/ttn/{bsu}"
    #print(f"Message with st=15 pddddd to -----")
    result = client.publish(new_topic,
json.dumps(new_payload).encode('utf-8'), qos=MQTT_QOS,
retain=False)
    status = result.rc
    if status == 0:
        print(f"Message with st=15 published to
{new_topic} -----")
    else:
        print(f"Failed to send message with st=15 to
{new_topic}, BSU result code: {status}")

if code == 13:
    bsu = data.get("bsu", None)
    if bsu:
        new_payload = json.loads(payload)
        new_payload['st'] = 13
        new_topic = f"grand_beton/ttn/{bsu}"
        result = client.publish(new_topic,
json.dumps(new_payload).encode('utf-8'), qos=MQTT_QOS,
retain=False)
        status = result.rc
        if status == 0:
            print(f"Message with st=13 published to
{new_topic}")
        else:
            print(f"Failed to send message with st=13 to
{new_topic}, result code: {status}")

elif "report" in topic:
    product_list = data.get("product", [])

```

```

current_loop_list = data.get("current_loop", [])
idPlant = data.get("bsu")

for product in product_list:
    dateStart = data.get("datetime") or "2025-01-01
00:00:00"
    timeEnd = product.get("time_done") or "2025-01-01
00:00:00" # Заменяем null на пустую строку
    vProduct = product.get("v_product") or 0 # Заменяем
null на 0
    loopNumber = product.get("num_loop") or 0 # Заменяем
null на 0
    vLoop = product.get("v_loop") or 0 # Заменяем null на
0
    driver = product.get("driver") or "" # Заменяем null
на пустую строку
    car = product.get("car") or "" # Заменяем null на
пустую строку
    classRecipe = product.get("class_recipe") or "" #
Заменяем null на пустую строку
    nameRecipe = product.get("name_recipe") or "" #
Заменяем null на пустую строку
    recipe = product.get("recipe") or "" # Заменяем null
на пустую строку
    idTtn = product.get("ttn") or 0 # Заменяем null на 0
(если это числовое поле)
    timeStart = product.get("time_start") or "2025-01-01
00:00:00" # Заменяем null на пустую строку
    num_loop = product.get("num_loop") or 0 # Заменяем
null на 0
    indProduct = product.get("ind_product") or 0 #
Заменяем null на 0

    ensure_connection()
    if dateStart:
        naive_date =
timezone.datetime.strptime(dateStart, "%Y-%m-%d %H:%M:%S")
        dateStart = timezone.make_aware(naive_date,
timezone.get_current_timezone())
        if not Product.objects.filter(idPlant=idPlant,
indProduct=indProduct).exists():
            product_obj = Product(
                dateStart=dateStart, timeEnd=timeEnd,
vProduct=vProduct, loopNumber=loopNumber,
                vLoop=vLoop, driver=driver, car=car,
classRecipe=classRecipe, nameRecipe=nameRecipe,
                recipe=recipe, idTtn=idTtn,
timeStart=timeStart, num_loop=num_loop, idPlant=idPlant,
                indProduct=indProduct
            )

```

```

        product_obj.save()

        for loop in current_loop_list:
            if loop.get("ind_product", None) ==
product.get("ind_product", None):
                ensure_connection()
                Reportcurrentloop.objects.create(
                    vLoop=loop.get("v_loop", None),
                    loopNumber=loop.get("num_loop", None),
                    code=loop.get("code", None),
                    dispenser=loop.get("dispenser", None),
                    doisingError=loop.get("err_dosing",
None),
doisingErrorPersent=loop.get("err_dosing_persent", None),
                    doisingKorr=loop.get("korr_dosing",
None),
                    humidityKorr=loop.get("korr_humidity",
None),
weightFactLoop=loop.get("weight_fact_loop", None),
                    weightFactM3=loop.get("weight_fact_m3",
None),
weightRecipeLoop=loop.get("weight_recipe_loop", None),
weightRecipeM3=loop.get("weight_recipe_m3", None),
                    idProduct=product_obj.id,      # Передаем
идентификатор Product
                    indProduct=indProduct
                )

                ensure_connection()
                #Report.objects.create(json=payload)

# Callback функция при получении сообщения
def on_message(client, userdata, msg):
    print(f"Received message:-----otladka")
    save_message(client, msg.topic, msg.payload.decode())

    #result = client.publish(MQTT_TOPIC_PUB,
msg.payload.decode('utf-8').encode('utf-8'), qos=MQTT_QOS)
    #status = result.rc
    #if status == 0:
        #print(f"Message published to {MQTT_TOPIC_PUB}")
    #else:
        #print(f"Failed to send message to {MQTT_TOPIC_PUB},
result code: {status}")

```

```
# Callback функция при подключении к брокеру
def on_connect(client, userdata, flags, rc):
    print(f"Connected to MQTT broker with result code {rc}")
    client.subscribe(MQTT_TOPICS_SUB)

def start_mqtt_client():
    client = mqtt.Client(client_id=MQTT_CLIENT_ID)
    client.on_connect = on_connect
    client.on_message = on_message
    client.username_pw_set(MQTT_USERNAME, MQTT_PASSWORD)
    client.connect(MQTT_BROKER, MQTT_PORT, 60)
    client.loop_start()
    return client

lock_file = '/tmp/mqtt_client.lock'

if os.path.exists(lock_file):
    print("MQTT client is already running.")
    sys.exit()

# Создание файла блокировки
with open(lock_file, 'w') as f:
    f.write(str(os.getpid()))

try:
    # Ваш код для старта MQTT клиента
    start_mqtt_client()
finally:
    # Удаление файла блокировки при завершении работы
    os.remove(lock_file)
```

ДОДАТОК В
Демонстраційний матеріал

