



**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**Факултет „Компютърни системи и технологии“**

**Катедра „Компютърни системи“**

**маг. инж. Венцислав Бойков Начев**

**ПРОЕКТИРАНЕ И ИЗСЛЕДВАНЕ НА ПЛАТФОРМИ ЗА  
ОТДАЛЕЧЕНО ОБУЧЕНИЕ ПО ВГРАДЕНИ  
УПРАВЛЯВАЩИ СИСТЕМИ И РОБОТИКА**

**А В Т О Р Е Ф Е Р А Т**

на дисертация за придобиване на образователна и научна степен  
**"ДОКТОР"**

Област: 5. Технически науки

Професионално направление: 5.3 Комуникационна и компютърна техника

Научна специалност: Системно програмиране

**Научен ръководител: проф. д-р Даниела Гоцева**

СОФИЯ, 2024 г.

Дисертационният труд е обсъден и насочен за защита от Катедрения съвет на катедра „Компютърни системи“ към Факултет „Компютърни системи и технологии“ на ТУ-София на редовно заседание, проведено на 16.12.2024г. .

Публичната защита на дисертационния труд ще се състои на 24.03.2025 г. от 13:00 часа в Конферентната зала на БИЦ на Технически университет – София на открито заседание на научното жури, определено със заповед №ОЖ-5.3-02/ 08.01.2025 г. на Ректора на ТУ-София в състав:

1. проф. д-р инж. Румен Трифонов – председател
2. проф. д-р инж. Милена Лазарова – научен секретар
3. проф. д-р инж. Станислав Симеонов
4. проф. д-р инж. Нина Синягина
5. проф. д-р инж. Александър Бежарски

Рецензенти:

1. проф. д-р инж. Нина Синягина
2. проф. д-р инж. Станислав Симеонов

Материалите по защитата са на разположение на интересуващите се в канцеларията на Факултет „Компютърни системи и технологии“ на ТУ-София, блок №1, кабинет №1431.

Дисертантът е задочен докторант към катедра „Компютърни системи“ на факултет „Компютърни системи и технологии“ . Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. инж. Венцислав Бойков Начев

Заглавие: **Проектиране и изследване на платформи за отдалечено обучение по вградени управляващи системи и роботика**

Тираж: 30 броя

Отпечатано в ИПК на Технически университет – София

# **I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД <sup>1</sup>**

---

## **Актуалност на проблема**

В условията на динамично развиващите се технологии и нарастващата автоматизация във всички сфери на индустрията, обучението по вградени управляващи системи и роботика придобива все по-голямо значение за подготовката на квалифицирани специалисти. Същевременно образователните институции се сблъскват със значителни предизвикателства при осигуряването на качествено практическо обучение, включително ограничен достъп до специализирано оборудване, недостатъчен брой лабораторни установки спрямо броя обучаеми и необходимост от физическо присъствие за работа със специализирана апаратура. Традиционният модел на обучение, изискващ директен достъп до лабораторно оборудване, се оказва все по-неефективен в контекста на съвременните образователни нужди, особено при необходимост от дистанционно или хибридно обучение. Освен това, високата стойност на роботизираните системи и специализираното оборудване често прави невъзможно за отделните образователни институции да осигурят достатъчен брой работни места за всички обучаеми, което допълнително ограничава възможностите за придобиване на практически опит и развитие на необходимите професионални умения.

## **Цел на дисертационния труд, основни задачи и методи за изследване**

Основната цел на дисертационния труд е проектирането, разработването и изследването на платформи за отдалечено обучение по вградени управляващи системи и роботика.

За постигане на поставената цел са формулирани следните основни задачи:

**Задача 1:** Проектиране на платформа за отдалечено програмиране на вградени управляващи системи и работи, което включва задълбочен анализ на съвременните технологични решения и подходи за нейната реализация. В рамките на тази задача се извършва проучване на наличните технологии, разработват се архитектурни концепции за разпределени системи, дефинират се изисквания към отделните компоненти.

**Задача 2:** Разработка на тестова платформа за отдалечено програмиране на вградени управляващи системи и работи. Това включва имплементацията на всички софтуерни модули, интеграцията на различни типове хардуерни устройства, създаването на интуитивни потребителски

интерфейси и осигуряването на надеждна и сигурна комуникация между отделните компоненти на системата.

**Задача 3:** Изследване на тестовата платформа за отдалечено програмиране на вградени управляващи системи и роботи в образователна среда. В рамките на тази задача се разработват методики за интегрирани уроци, провеждат се тестови занятия с различни групи обучаеми, анализира се ефективността на платформата в учебния процес и се оценяват педагогическите аспекти и постигнатите образователни резултати

### **Научна новост**

Научната новост на дисертационния труд се състои в разработването на иновативен подход за създаване на модулни платформи за отдалечено програмиране на вградени управляващи системи и роботи в образователен контекст. Разработена е цялостна архитектура, която интегрира разнородни роботизирани системи и контролери в единна образователна платформа, като същевременно осигурява:

1. Нова концепция за изграждане на разпределени лаборатории, базирана на модулна архитектура, която позволява лесно разширяване и интеграция на различни типове устройства;
2. Оригинален подход за емуляция на физически устройства, който дава възможност за пълноценна работа с лабораторното оборудване в дистанционен режим;
3. Иновативно решение за автоматизирано управление на програмирането и комуникацията с роботизирани системи чрез уеб-базиран интерфейс;
4. Нова методика за провеждане на практически занятия по програмиране на вградени системи и роботи в дистанционен формат, която успешно съчетава теоретичното обучение с придобиването на практически умения.

### **Практическа приложимост**

Предложената в дисертационния труд архитектура създава солидна основа за реализиране на платформи за отдалечено програмиране на вградени управляващи системи и роботи, като предоставя цялостна рамка за тяхното проектиране и имплементация. Разработената платформа за отдалечено програмиране на вградени управляващи системи и роботи демонстрира широк потенциал за приложение в различни образователни и професионални среди. В сферата на средното образование, системата може успешно да се интегрира в учебния процес на професионалните и профилираните гимназии, където да подпомага обучението по програмиране, роботика и автоматизация. На ниво висше образование, платформата

предоставя ценен инструмент за техническите университети при провеждането на лабораторни упражнения и научни изследвания в областта на вградените системи и роботиката. Извън образователния контекст, системата намира приложение в научноизследователски институти за провеждане на експерименти с отдалечен достъп до специализирано оборудване. Особено важно е практическото приложение на платформата в технологични компании, където може да се използва за дистанционно програмиране и управление на индустриални контролери и роботизирани системи, като по този начин оптимизира работните процеси и намалява необходимостта от физическо присъствие при програмиране и поддръжка на оборудването. Универсалността и модулната архитектура на системата я правят подходяща за адаптиране към специфичните нужди на всяка организация, независимо дали става въпрос за образователни, научни или индустриални приложения.

### **Апробация**

В рамките на дисертационния труд е разработена и успешно тествана експериментална платформа за отдалечено програмиране на вградени управляващи системи и роботи. Апробацията на системата е проведена с ученици от НПП по КТС - гр. Правец, обучаващи се в специалност "Системно програмиране", както и с ученици, изучаващи предмета "Въведение в операционните системи и вградените системи" от Националната програма "Обучение за ИТ кариера и умения". Практическото тестване на платформата в реална образователна среда убедително демонстрира приложимостта на предложената архитектура и потвърждава нейната ефективност за провеждане на дистанционно обучение по програмиране на вградени системи и роботи.

### **Публикации**

Резултатите от дисертационния труд са представени в шест научни публикации, от които две са самостоятелни и три са в съавторство с научния ръководител. Четири публикации индексирани в базата данни Scopus.

### **Структура и обем на дисертационния труд**

Дисертационният труд е в обем от **200** страници, като включва увод, **4** глави за решаване на формулираните основни задачи, списък на основните приноси, списък на публикациите по дисертацията и използвана литература. Цитирани са общо **109** литературни източници, като **96** са на латиница и **5** на кирилица, а останалите са интернет адреси. Работата включва общо **99** фигури и **13** таблици. Номерата на фигурите и таблиците в автореферата съответстват на тези в дисертационния труд.

## II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

### ГЛАВА 1. СЪВРЕМЕНО СЪСТОЯНИЕ НА ПЛАТФОРМИТЕ ЗА ОТДАЛЕЧЕНО ОБУЧЕНИЕ ПО ВГРАДЕНИ УПРАВЛЯВАЩИ СИСТЕМИ И РОБОТИКА

Платформите за отдалечено обучение по вградени системи и роботика позволяват на ученици и студенти достъп до реално лабораторно оборудване чрез лични устройства, използвайки само уеб браузър и интернет. Тези системи осигуряват двупосочна комуникация, която позволява както изпращане на команди към устройствата, така и получаване на обратна връзка в реално време, създавайки условия за ефективно практическо обучение. Хибридният модел, съчетаващ присъствено и онлайн обучение, включва видеонаблюдение, интерактивни интерфейси и интеграция с платформи за управление на обучението, като оптимизира ресурсите и улеснява планирането и организацията на учебния процес.

#### Обзор на съществуващи платформи за отдалечено обучение

В рамките на направеното проучване, обобщено в Таблица 1, са анализирани единадесет водещи платформи за отдалечено обучение от различни страни по света, включително Испания, Швеция, Мексико, Бразилия, Норвегия, Австралия, Русия и Сърбия. Разгледаните платформи обхващат широк спектър от дисциплини, включително електроника, вградени системи, роботика, компютърно зрение и управляващи системи. Сравнителен анализ между разгледаните отдалечени лаборатории е направен в таблица 1.

Таблица 1 Сравнителен анализ на отдалечени лаборатории

Платформа	Поддържани дисциплини	Потребителски и интерфейс	Гъвкавост	Целева аудитория	Държава
<b>WebLab Deusto</b>	Физика, Електроника, Химия	Уеб-базиран	Висока (Отворен код, персонализиращ се)	Университети, Изследователски институти	Испания
<b>Elab</b>	Електроника	Уеб-базиран, В реално време	Средна (Ограничена до електроника)	Университети, Технически училища	Мексико
<b>RELLE</b>	Вградени системи, Електроника	Уеб-базиран, Интеграция с Arduino	Средна (Фокус върху вградени системи)	Средни училища, Университети	Испания
<b>VISIR</b>	Електроника	Уеб-базиран, Достъп до реални инструменти	Висока (Поддържа реални инструменти)	Университети, Научни институти	Швеция
<b>RExLab</b>	Електроника, Вградени системи	Уеб-базиран, Интерактивен	Средна (Част от университетска мрежа)	Университети, Изследователи	Бразилия

Платформа	Поддържани дисциплини	Потребителски и интерфейс	Гъвкавост	Целева аудитория	Държава
<b>ACEL</b>	Управляващи системи	Уеб-базиран, Интеграция с LabVIEW	Ниска (Специализирана за управление)	Технически университети	Сърбия
<b>RECOLA B</b>	Управляващи системи	Уеб-базиран, Интеграция с MATLAB/Simulink	Средна (Изисква MATLAB/Simulink)	Университети, Изследователски центрове	Испания
<b>AIM-LAB</b>	Електроника	Уеб-базиран, Интеграция с GPIB	Ниска (Специализирана за измервания)	Изследователски институти, Университети	Норвегия
<b>NetLab</b>	Програмиране на миконтролери	Графичен интерфейс, Симулирани лабораторни инструменти	Висока (Различни инструменти и симулации)	Университети, Средни училища	Австралия
<b>uCV-Lab</b>	Компютърно зрение, Роботика	Уеб-базиран, Интерактивен	Средна (Фокус върху компютърно зрение и роботика)	Университети, Изследователски институти	Русия
<b>Remote Robotics Lab UA</b>	Роботика	Уеб-базиран, Системи за дистанционно управление	Средна (Фокус върху роботиката)	Университети, Средни училища	Испания

## ГЛАВА 2. ПРОЕКТИРАНЕ НА ПЛАТФОРМА ЗА ОТДАЛЧЕНО ПРОГРАМИРАНЕ НА ВГРАДЕНИ УПРАВЛЯВАЩИ СИСТЕМИ И РОБОТИ

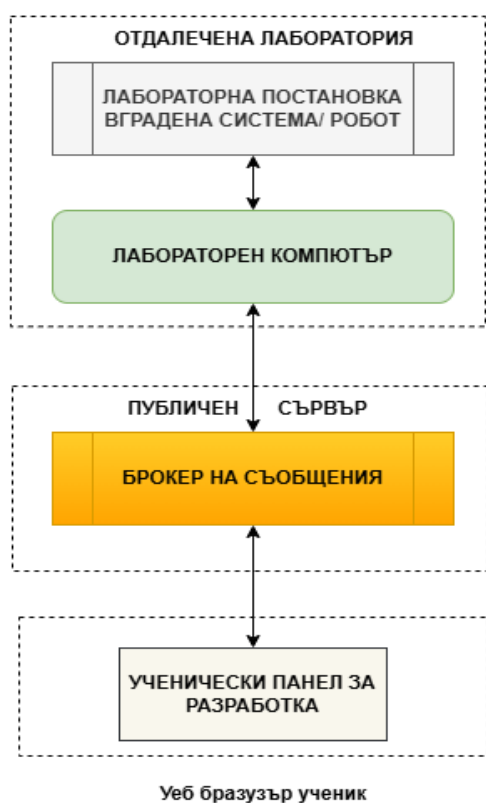
В тази глава се разглежда проектирането и анализа на технологии за реализацията на модулна платформа за отдалечено програмиране както на вградени управляващи системи, така и на образователни работи и индустриални машини. За реализацията се използва разпространен и достъпен хардуер и софтуерни системи с отворен код. Основният принцип на платформата е модулният дизайн и архитектура, която включва множество софтуерни и хардуерни компоненти, работещи в синхрон. Целта е системата да бъде достъпна и лесно мултиплицирана, което ще позволи на повече потребители да я прилагат в образователния процес и в практическите си проекти.

### Архитектура на платформа за отдалечено програмиране на вградени управляващи системи.

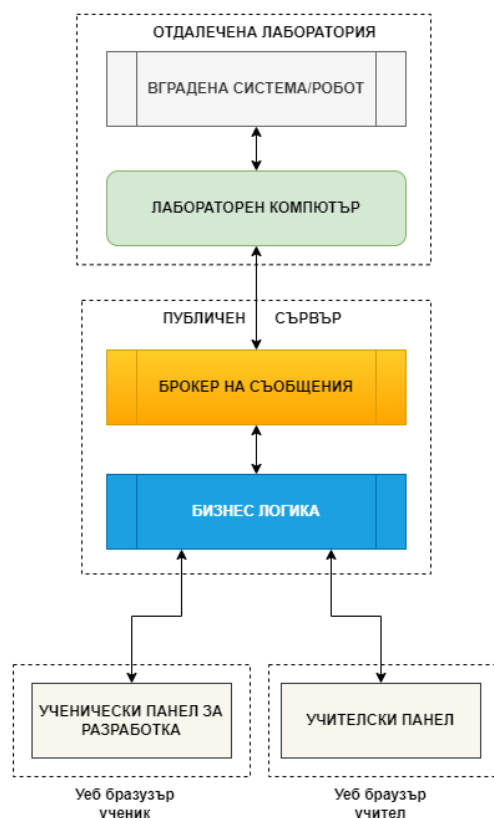
Архитектурата на платформата е разделена на четири основни компонента и е представена на фигура 21:

- **Уеб базирана среда за разработка на кода, управление и наблюдение на експеримента (Web-based IDE)**
- **Брокер на съобщения (Message broker)**
- **Лабораторен компютър (Laboratory Computer)**
- **Лабораторна постановка – Вградена система и/или роботизиранз система (Laboratory equipment)**

При вторият вариант за реализация, представен на фигура 22, на системата включва добавянето на допълнителна сървърна услуга, която да функционира като посредник между брокера на съобщенията и брауъра на компютъра. Тази архитектурна конфигурация може да осигури повишена сигурност на системата, като ограничи директната комуникация между клиентите и брокера на съобщения. Освен това, внедряването на такъв междинен слой би позволило създаването на допълнителен учителски интерфейс, който да предоставя администриране на ученическите проекти и управление на ресурсите. Тази опция е най-подходяща в случаи, когато системата трябва да отговаря на специфични изисквания или стандарти за сигурност и управление.



*Фигура 21 Архитектура на система за отдалечено програмиране на вградени управляващи системи*



*Фигура 22 Архитектура на системата с добавен учителски интерфейс*



## Формално описание на системата:

- Ресурси (хардуерни и софтуерни):  $R = (id, W, FUNC)$ ,
- Услуги:  $S = (id, Ra, Rr, F)$
- Интерфейси:  $I = (Sy, D)$ , Протоколи:  $P = (I_s, I_t, \Sigma)$ ,
- Слоеве:  $L = \{L_i \mid 0 \leq i \leq n - 1\}$ 
  - Слой 0 ( $L_0$ ): Лабораторни експерименти.
  - Слой 1 ( $L_1$ ): Междинни услуги, включително компилация, емуляция, активност на устройства.
  - Слой 2 ( $L_2$ ): Брокери на съобщения
  - Слой 3 ( $L_3$ ): Потребителски уеб интерфейс.
- Релация между горен и долен слой:  $L_{i+1} \nabla_d L_i$
- Релация между долен и горен слой:  $L_i \nabla_s L_{i+1}$
- Комуникация от потребител към лабораторен експеримент:  $L_3 \nabla_d L_2 \nabla_d L_1 \nabla_d L_0$
- Обратна връзка:  $L_0 \nabla_s L_1 \nabla_s L_2 \nabla_s L_3$

## Избор на брокер на съобщения

Брокерите на съобщения (Message Brokers) са софтуерни компоненти, които действат като междинен слой (middleware) в разпределените системи, позволявайки на различни приложения, услуги и системи да комуникират помежду си чрез обмен на съобщения. Проучени са четири технологии, с които може да се реализира брокера на съобщения.

В контекста на реализацията на платформа за отдалечено програмиране на вградени системи и роботи, всяка от разгледаните технологии - MQTT, Redis, RabbitMQ и Apache Kafka, предоставя възможност за изграждане на надеждна комуникационна инфраструктура. Изборът на конкретна технология зависи от специфичните изисквания на проекта като брой едновременни потребители, необходимост от гарантирана доставка на съобщения, сложност на маршрутизацията, изисквания за латентност, мащабируемост на системата и наличните хардуерни ресурси. Сравнение на четирите технологии е представено в таблица 2.

**Таблица 2 Сравнителна характеристика между брокери на съобщения**

Характеристика	MQTT	Redis	RabbitMQ	Apache Kafka
Модел на съобщенията	Publish/Subscribe	Pub/Sub + Queue	Pub/Sub + Queue	Pub/Sub + Log
Протокол	MQTT	Redis Protocol	AMQP, MQTT, STOMP	Kafka Protocol
Максимален размер на съобщението	Неограничен	512MB	Неограничен	1MB
Гарантирана доставка	QoS 0,1,2	Не	Да	Да
Непрекъснатост	Опционално	Опционално	Да	Да
Скалируемост	Средна	Висока	Средна	Много висока

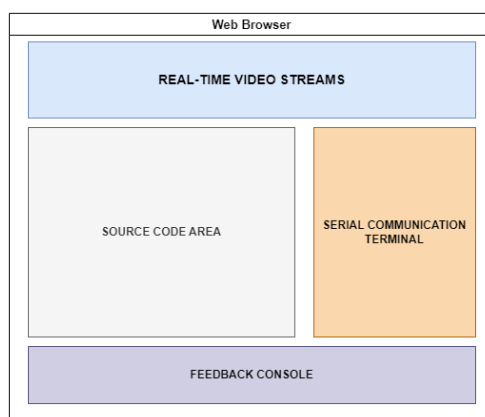
Характеристика	MQTT	Redis	RabbitMQ	Apache Kafka
Производителност (msg/sec)	~100К	~100К	~20К	~1М
Латентност	Много ниска	Много ниска	Ниска	Средна
Използване на памет	Ниско	Средно	Средно	Високо
Сложност на конфигурацията	Ниска	Ниска	Средна	Висока

## Избор на лабораторен компютър

Лабораторният компютър трябва да представлява компактна компютърна система с многозадачна операционна система, която да позволява едновременното изпълнение на множество процеси и програми. От съществено значение е наличието на разнообразни интерфейси за свързаност, включително мрежови интерфейси за отдалечена комуникация и USB портове за свързване на периферни устройства. Критичен компонент за лабораторната система е наличието на GPIO (General Purpose Input/Output) интерфейс, който позволява директно управление и мониторинг на различни физически компоненти в лабораторните експерименти, като сензори, актуатори и други електронни елементи..

## Разработка на уеб базирана среда за разработка на кода, управление и наблюдение на експеримента (Web-based IDE).

Целта на този модул е да се използва от крайните потребители без да се изисква инсталиране и конфигуриране на какъвто и да е софтуер на техните компютри. Това позволява по-лесна поддръжка на системата и нейното разширяване и подобряване без да се зависи от крайните потребители. Структурата на уеб базираната среда за разработка е показана на фигура 31 и е разделена на 4 секции.



Фигура 31 Структура на уеб базираната среда за разработка

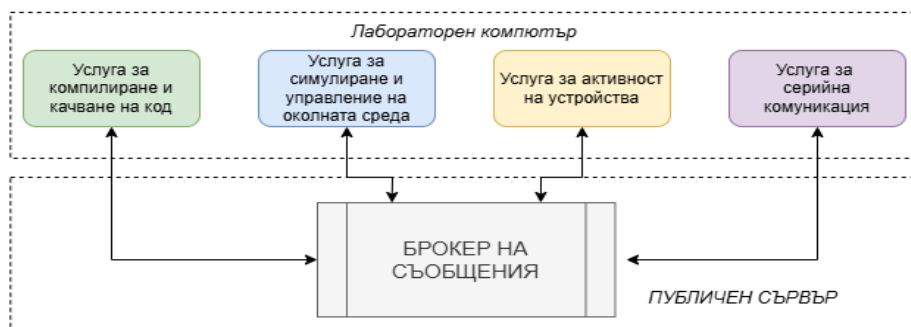
- Секция за редактор на програмен код (Source code Area):
- Секция за сериен терминал (Serial Communication Terminal):

- Секция за обратна връзка от процеса на програмиране и качване на кода (Feedback console):
- Секция за видеопредаване в реално време от отдалечената лаборатория:

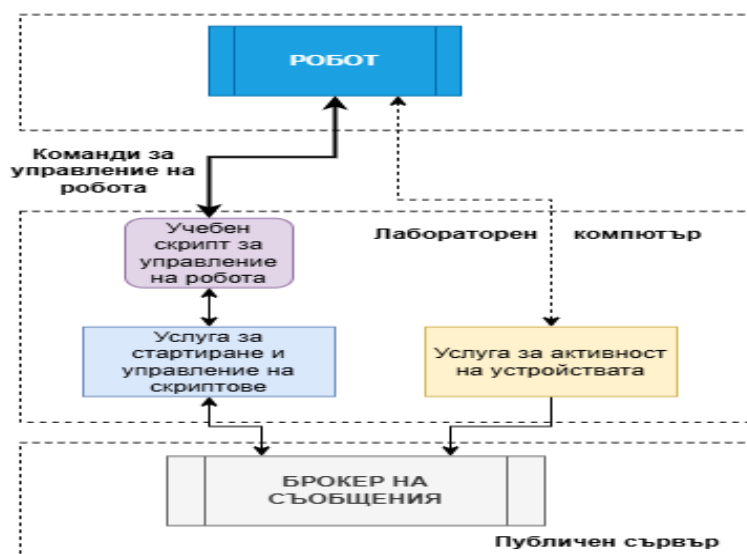
### Проектиране на работещите услуги на лабораторния компютър :

На лабораторната компютърна система са проектирани да изпълняват задачата пет основни услуги, всяка от които изпълнява специфична функция. На фигура 34 и фигура 38 са представни диаграми на изпълняваните услуги в лабораторния компютър.

- Услуга за компилиране и качване на код
- Услуга за емулиране и управление на заобикалящата среда на лабораторната постановка
- Услуга за активност на устройствата - „Heartbeat“
- Услуга за серийна комуникация
- Услуга за стартиране на скриптове за управление на работи



Фигура 34 Услуги на лабораторния компютър при програмиране на вградени управляващи системи



Фигура 38 Услуги на лабораторния компютър при програмиране на работи

## **Изводи**

1. Проектирането на платформа за отдалечено програмиране на вградени системи и роботи изисква комплексен подход, който да отговаря на множество технически и образователни изисквания.
2. Сравнителният анализ на различни брокери на съобщения (MQTT, Redis, RabbitMQ, Apache Kafka) показва, че MQTT протоколът е най-подходящ за реализацията на такава платформа
3. За лабораторен компютър е избран Raspberry Pi 4, който предлага оптимално съчетание от параметри.
4. Проектираната система от услуги на лабораторния компютър осигурява надеждно: Компилиране и качване на код; Симулиране на въздействия; Мониторинг на активността на устройствата; Серийна комуникация; Управление на роботизирани системи
5. Избраният набор от лабораторно оборудване (Arduino/ESP32 контролери, Misty II, Dobot Magician, DJI Tello) създава разнообразна образователна среда.
6. Анализът на технологиите за разработка на уеб базирания интерфейс показва, че традиционният стек (HTML/CSS/JavaScript) е достатъчен за реализация на базова функционалност, но за по-сложни интерфейси е препоръчително използването на модерни JavaScript рамки като React.js, Angular или Vue.js.
7. Предложената архитектура се отличава с това, че: обединява много технологии, гъвкава модулна архитектура, подходяща за разнообразни образователни и индустриални нужди.

## **ГЛАВА 3. РАЗРАБОТВАНЕ НА ПЛАТФОРМА ЗА ОТДАЛЧЕНО ПРОГРАМИРАНЕ НА ВГРАДЕНИ УПРАВЛЯВАЩИ СИСТЕМИ И РОБОТИ**

### **3.1 Общи положения**

В настоящата глава се представя проектирането и реализацията на иновативна образователна платформа, предназначена за отдалечено програмиране на вградени управляващи системи и роботизирани платформи. Фокусът на разработката е поставен върху създаването на гъвкава и достъпна среда за обучение, която успешно да отговори на съвременните предизвикателства в техническото образование. Имплементацията на платформата се базира на разпределената архитектура за отдалечен достъп до физически обекти, подробно описана в глава 2, като надгражда нейните възможности с специфични образователни функционалности и адаптирани интерфейси за учебни цели.

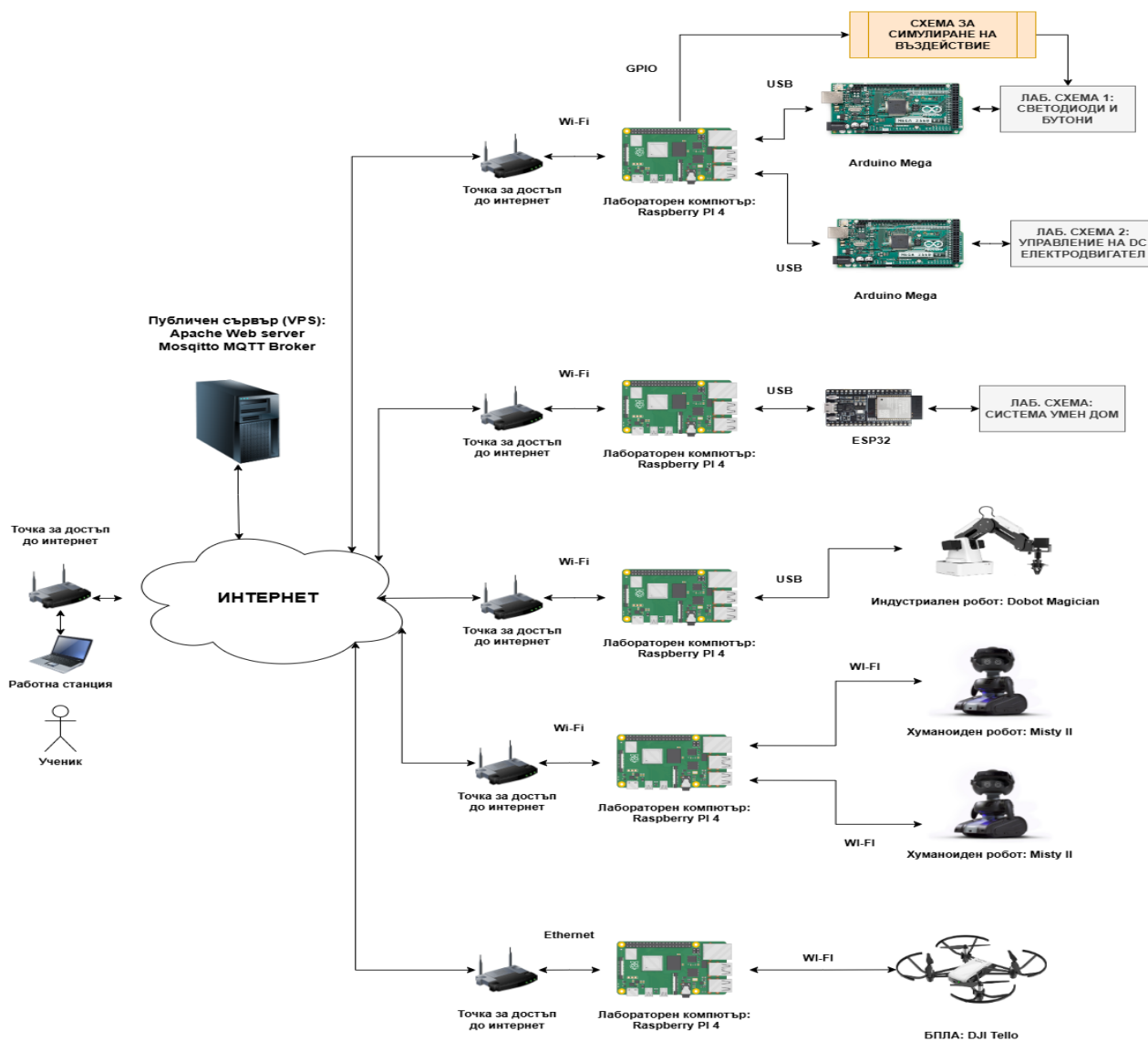
## 3.2 Архитектура и основни компоненти на платформата

### 3.2.1 Хардуерни компоненти на платформата

За реализацията на тестовата платформа като лабораторен компютър е избран Raspberry Pi 4.

1. **Arduino Mega 2560**
2. **ESP32 WROOM DevKit**
3. **Хуманоидни роботи Misty II**
4. **Dobot Magician**
5. **DJI Tello**

На фигура 44 е представена схема на апаратната реализация на платформата. На нея са представени лабораторните комплекси, свързани към лабораторните комоютри и средствата за комуникация с крайния потребител.



Фигура 44 Апаратна реализация на платформата

## Сървърна инфраструктура и комуникационен слой

За работата на платформата е използван публичен VPS (Virtual Private Server) сървър с инсталирана операционна система Debian 12 "Bookworm".

За осигуряване на надеждна и ефективна комуникация между компонентите на системата е избран протоколът MQTT. Неговите ключови характеристики са: Като конкретна имплементация на MQTT брокер е избран **Mosquitto MQTT** - популярно решение с отворен код, което се отличава със своята стабилност, лекота и ефективност. Mosquitto MQTT брокерът е инсталиран и конфигуриран на публичения сървър.

На същия VPS сървър е инсталиран и Apache HTTP Server - един от най-утвърдените уеб сървъри в световен мащаб. Apache хоства уеб-базирания потребителски интерфейс на платформата, през който учениците получават достъп до отдалечената лаборатория.

## Софтуерна реализация на платформата

### Уеб-базирана среда за разработка

За реализацията на уеб-базираното интегрирано средство за разработка (Web IDE) е използван утвърденият стек от уеб технологии: **HTML5**, **CSS3**, **JavaScript**. Този опростен подход към автентикацията е избран съзнателно, за да се улесни използването на системата в образователна среда, като същевременно се осигури необходимото ниво на сигурност и контрол над достъпа.

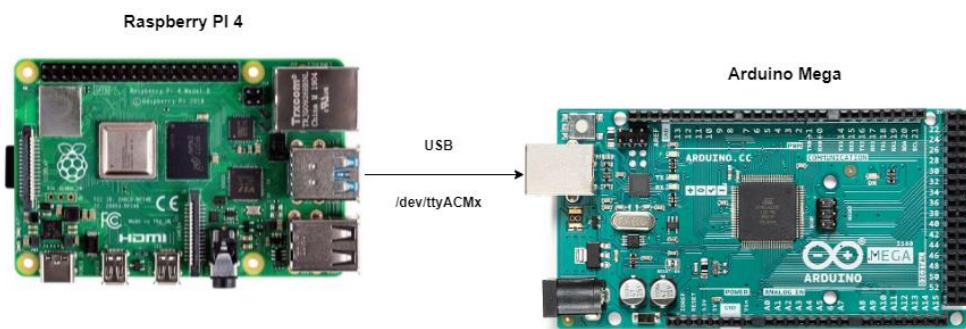
### Услуги в лабораторния компютър

Софтуерните услуги, изпълнявани на лабораторния компютър Raspberry Pi 4, са реализирани на езика Python.

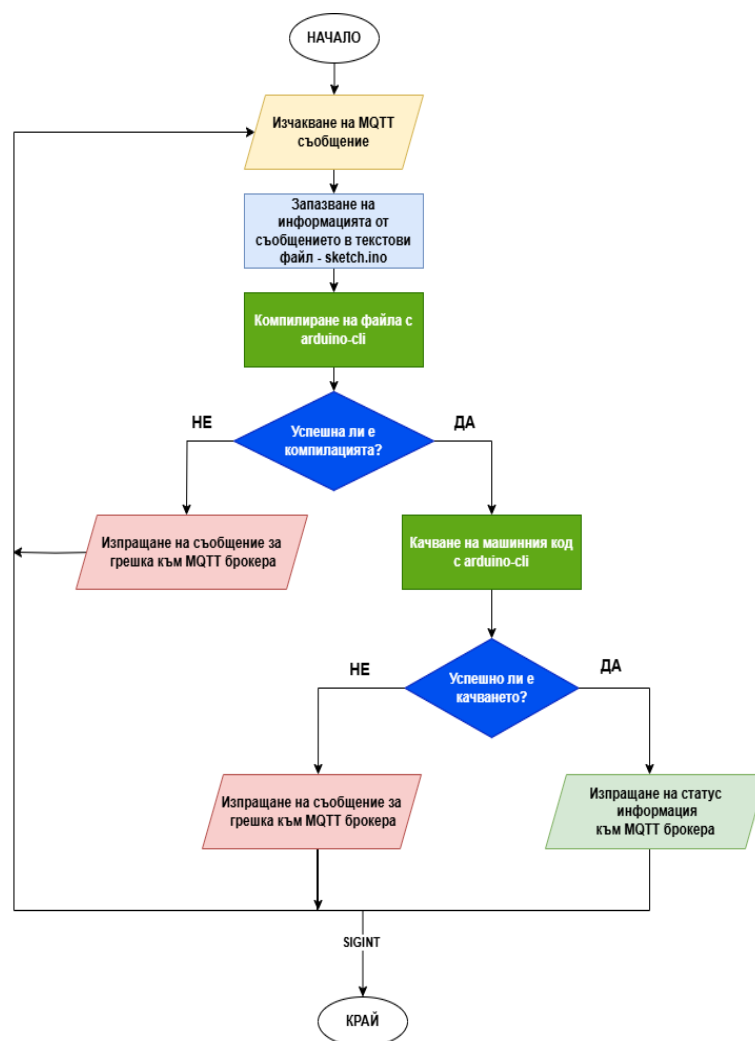
### Създаване на услуга за програмиране и качване на код за Arduino:

#### Услуга за отдалечено компилиране и качване на код

Чрез използването на услуга, написана на Python, се автоматизира процесът на програмиране на устройства като Arduino, като се извикват командите на **arduino-cli**. На фигура 48 е представена схемата на свързване между лабораторния компютър и ESP32, а на фигура 51 е представена блокова схема на алгоритъма.



Фигура 48 Схема на свързване между RaspberryPI и Arduino

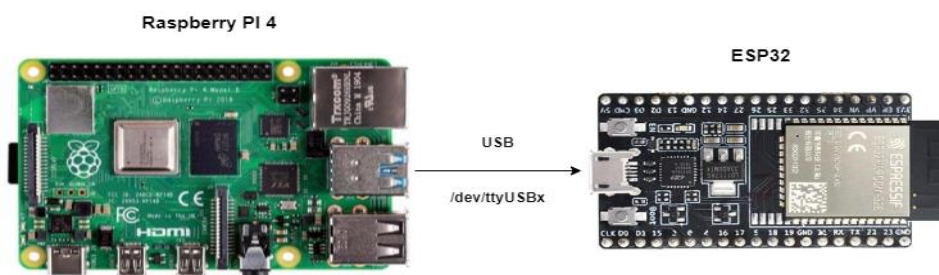


Фигура 51 Блокова схема на алгоритъм за автоматизирано програмиране на Arduino

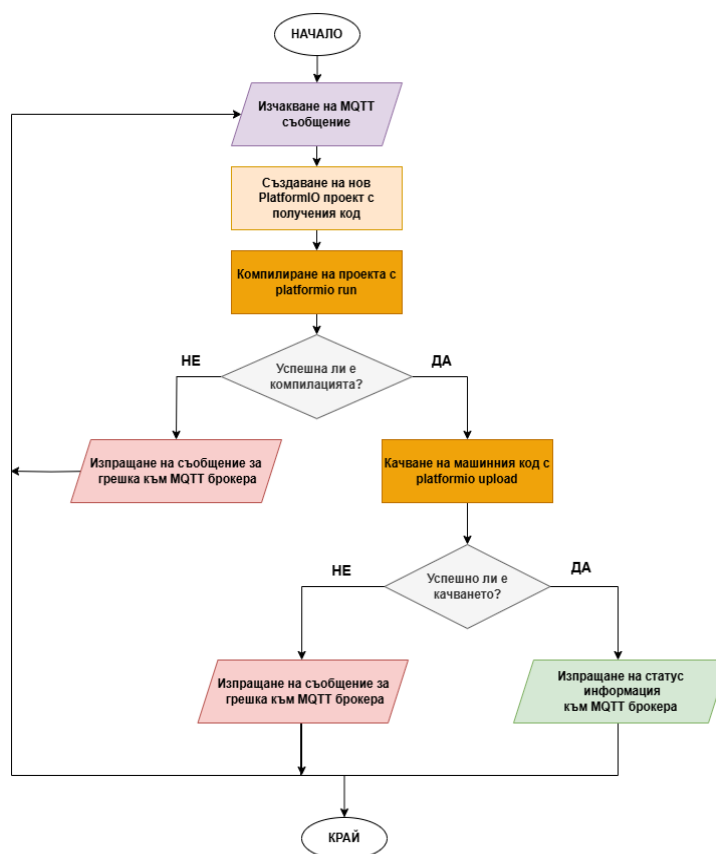
Програмата използва два основни модула: **paho-mqtt** (библиотека с отворен код, разработена от **Eclipse Foundation**) за комуникация чрез протокола MQTT и **subprocess** за изпълнение на системни команди, свързани с компилиране и качване на кода с помощта на **arduino-cli**.

## Създаване на услуга за програмиране и качване на код за ESP.

Програмирането на контролера **ESP32 WROOM DevKit** се извършва през сериен интерфейс и от Bootloader, по начин описан в предходната точка. За компилиране и качване на код в описаната система се използва **PlatformIO**. На фигура 52 е представена схемата на свързване между лабораторния компютър и ESP32, а на фигура 53 е представена блокова схема на алгоритъма.



Фигура 52 Свързване на RaspberryPI и ESP32



Фигура 53 Блокова схема на алгоритъм за автоматизирано програмиране на ESP32

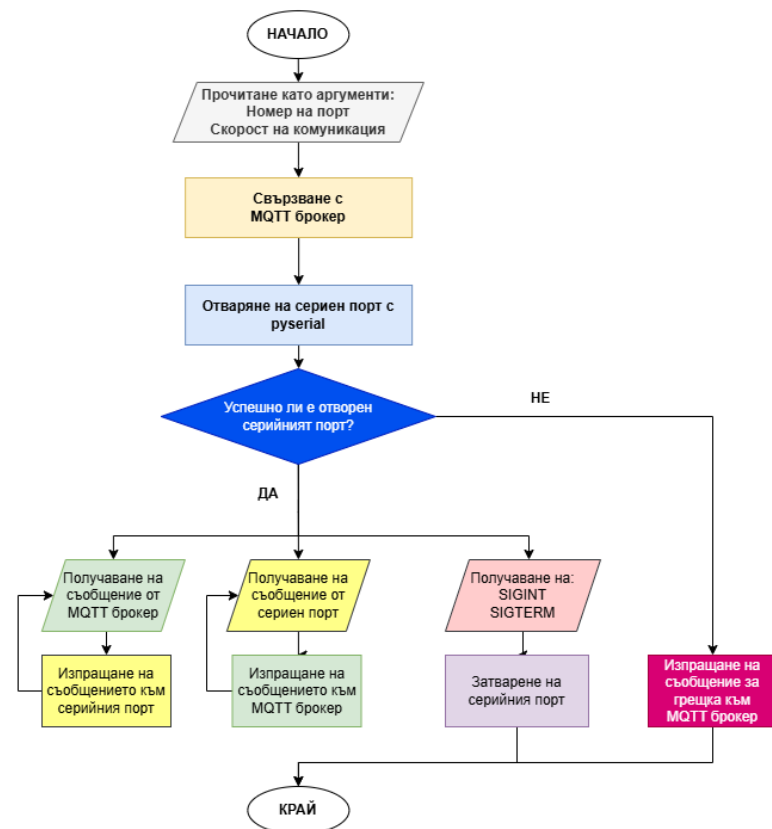
Докато с **arduino-cli** компилирането и качването се извършва чрез командите `arduino-cli compile` и `arduino-cli upload`, тук използваме командите



на PlatformIO: `platformio run` за компилиране и `platformio run --target upload` за качване. Това прави процеса много по-универсален, тъй като PlatformIO автоматично избира необходимите инструменти и зависимости за ESP32.

## Създаване на услуга за серийна комуникация

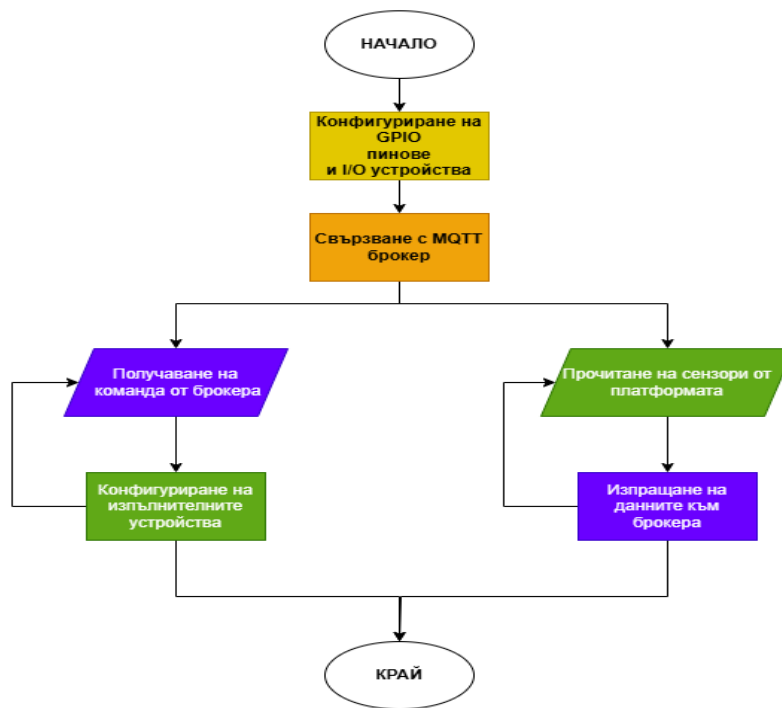
При разработката на програми за вградени системи от съществено значение е възможността за наблюдение на изходните данни и интерактивна комуникация с устройството. На фигура 54 е представена блокова схема на алгоритъма.



Фигура 54 Блокова схема на алгоритъм за серийна комуникация

## Създаване на услуга за контрол и емуляция на заобикалящата среда на лабораторната постановка

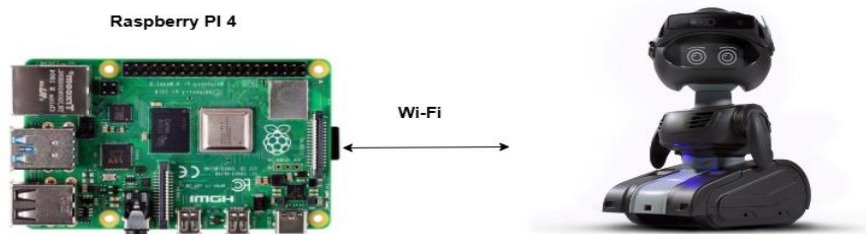
При разработката на програми за вградени системи често се изисква тестване и валидация на софтуера в различни условия на околната среда, което може да бъде предизвикателство, особено когато разработчиците нямат пряк физически достъп до лабораторната постановка. На фигура 56 е представена блокова схема на алгоритъма.



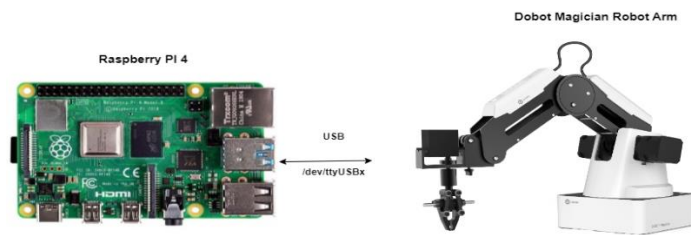
Фигура 56 Блокова схема на алгоритъм за емуляция на елементи

### Създаване на услуга за програмиране на работи

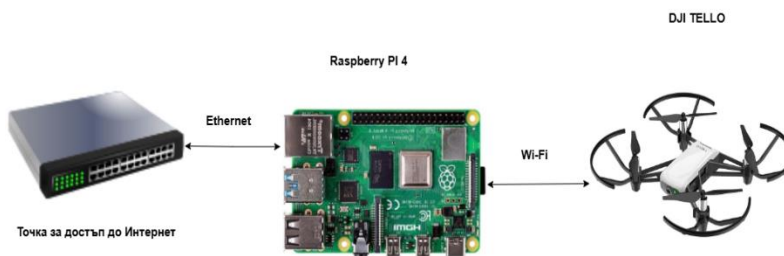
Услугата поддържа различни команди - изпълнение на програмен код, спиране на изпълнението и управление на специфични функции като камера. Всеки робот се идентифицира чрез уникален номер, който се използва както за рутване на MQTT съобщенията, така и за изолиране на изпълнението на програмите за различните работи. Програмният код за всеки робот се записва в отделен файл и се изпълнява в самостоятелен процес, което осигурява изолация и предотвратява конфликти между различните програми. На фигура 64 е представена блокова схема на алгоритъма. Подобен принцип на работа е приложен и при другите роботизирани платформи в системата. На фигура 58, 59 и 63 са представени схеми на свързване на лабораторен компютър и програмируема роботизирана система.



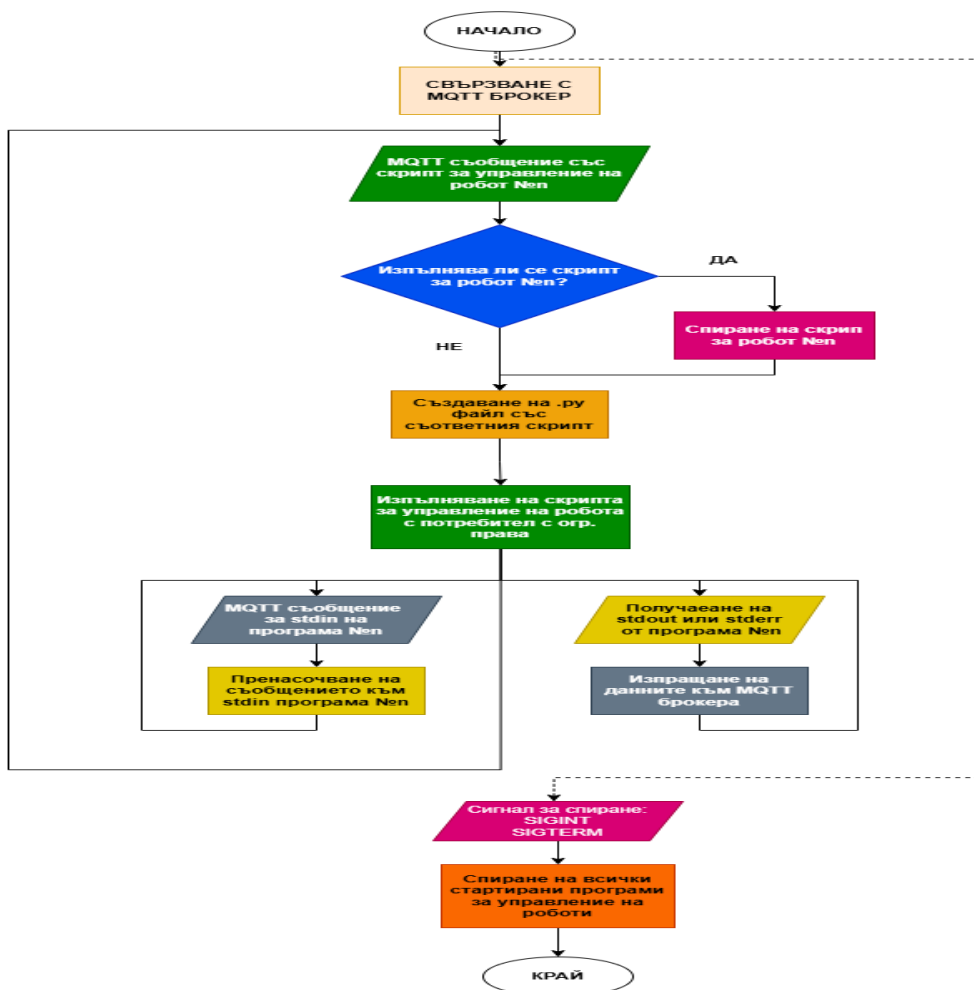
Фигура 58 Схема на свързване между RaspberryPI и Dobot magician



Фигура 59 Схема на свързване между RaspberryPI и Dobot magician



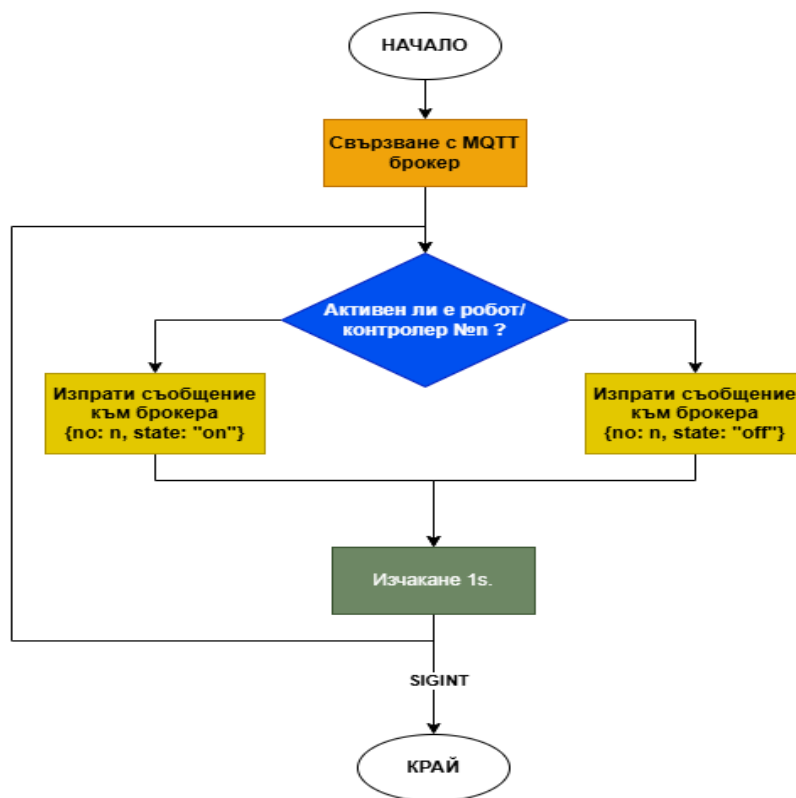
Фигура 63 Комуникация между DJI Tello и Raspberry PI



Фигура 64 Блокова схема на алгоритъм за програмиране на роботи

## Създаване на услуга за активност на устройствата - „Heartbeat“

В платформата е необходимо да бъде интегриран механизъм за **heartbeat** сигнал, който да информира крайния клиент за текущото състояние на лабораторията и наличността на свързаните Целта на услугата е да информира за активността на свързаните вградени системи и работи към съответния лабораторен компютър. На фигура 65 е представена блокова схема на алгоритъма.

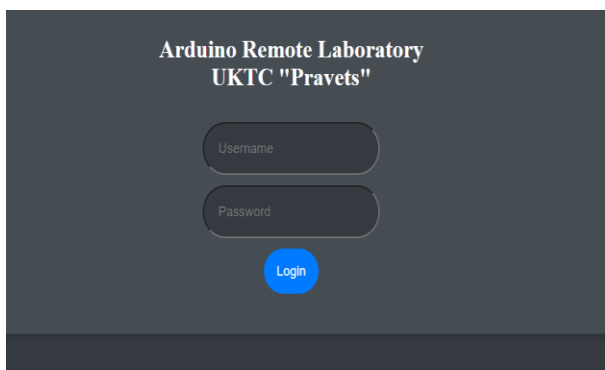


Фигура 65 Блокова схема на алгоритъм за активност на устройствата

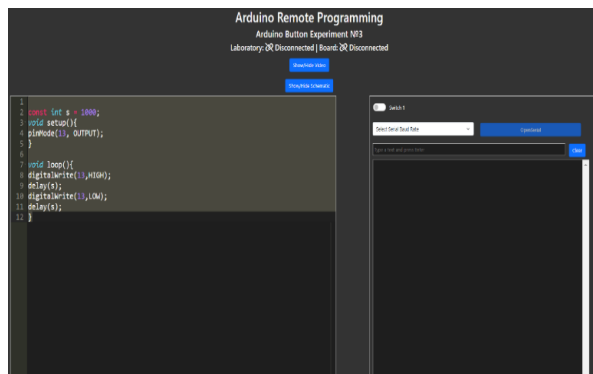
Тази унифицирана структура на услугата за проверка на активността осигурява надежден механизъм за мониторинг на свързаността с всички типове устройства в системата, независимо от специфичния начин на комуникация с тях.

**Създаване на уеб-базирана графична среда за програмиране, управление и мониторинг**

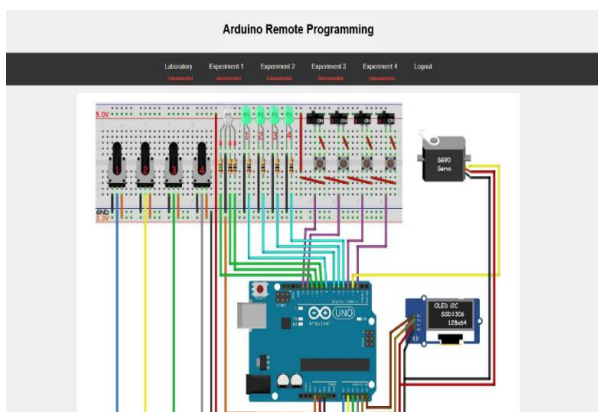
За реализацията на тестовата Web IDE система е използван традиционният стек от уеб технологии - HTML5 за структуриране на съдържанието, CSS3 за стилизиране и JavaScript за клиентска функционалност. От фигура 66 до фигура 69 са представени уеб-базирани графични интерфейси от разработената тестова платформа.



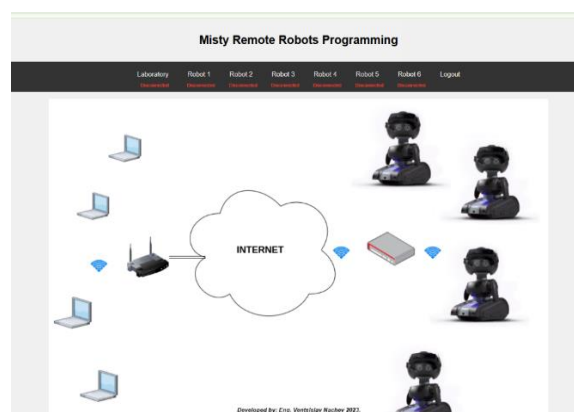
Фигура 66 Страница за вход



Фигура 67 Страница за програмиране на Arduino



Фигура 68 Начална страница на частта за програмиране на вградени управляващи системи



Фигура 69 Начална страница на частта за програмиране на робот

## Изводи

На базата на разработката и реализацията на платформата за отдалечено програмиране на вградени управляващи системи и роботи, могат да бъдат направени следните изводи:

1. **Реализация на архитектурата:** Успешно е имплементирана модулна архитектура, базирана на MQTT протокол за комуникация. Интегрирани са разнообразни хардуерни платформи в единна система. Постигната е висока надеждност и ефективност на комуникацията чрез

използването на Mosquitto MQTT брокер. Осигурена е гъвкавост при разширяване на системата с нови функционалности

2. **Имплементация на основните услуги:** Разработена е услуга за компилиране и качване на код, поддържаща различни микроконтролерни платформи (Arduino, ESP32); Създадена е услуга за симулация на заобикалящата среда чрез електромагнитни релета
3. **Реализация на роботизираните компоненти:** Успешно са интегрирани различни типове роботизирани системи (Misty II, Dobot Magician, DJI Tello); Разработени са специализирани модули за управление на всеки тип робот
4. **Потребителски интерфейс:** Създаден е интуитивен веб-базиран интерфейс за програмиране и управление
5. **Техническа реализация:** Използвани са съвременни технологии с отворен код;

Тези изводи потвърждават успешната реализация на поставените цели и създават основа за бъдещо развитие и усъвършенстване на платформата в образователен контекст.

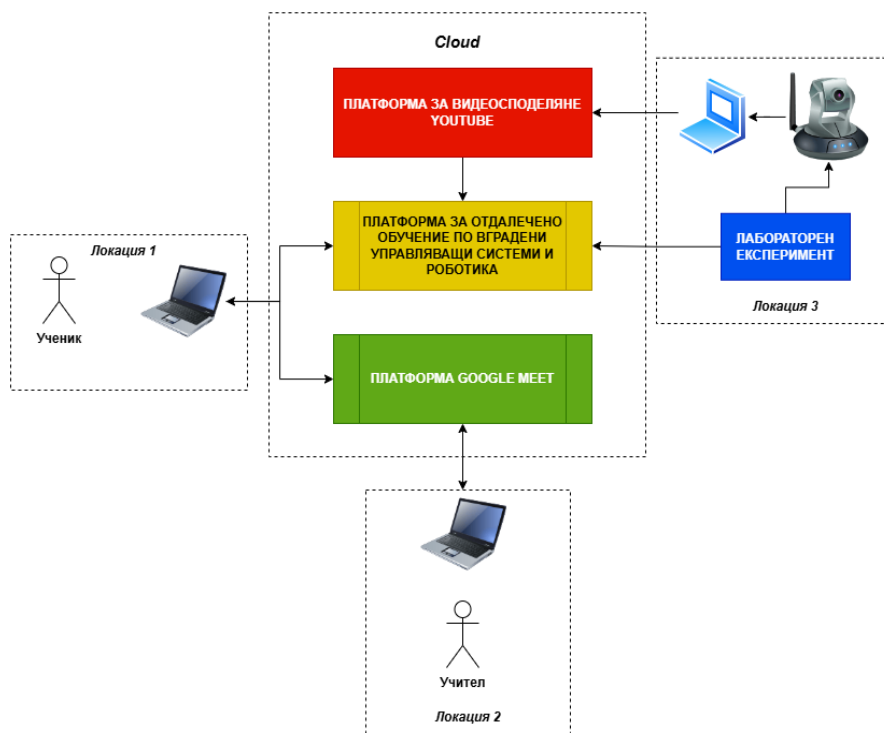
## **ГЛАВА 4. ИЗСЛЕДВАНЕ И ИЗПОЛЗВАНЕ НА ПЛАТФОРМИТЕ ЗА ОТДАЛЕЧЕНО ПРОГРАМИРАНЕ НА ВГРАДЕНИ УПРАВЛЯВАЩИ СИСТЕМИ И РОБОТИ В ОБРАЗОВАТЕЛНИЯ ПРОЦЕС**

В тази глава ще бъдат разгледани разработката на конкретни уроци и експерименти с цел тестване на разработена собствена платформа за отдалечено програмиране на вградени управляващи системи и роботи, базирана на архитектурата, описана в глава 2 и разработената платформа в глава 3.

В експериментите са участвали ученици от НПГ по КТС - гр. Правец, което предоставя реален контекст за прилагане на разработената платформа

### **4.1 Опитна постановка**

За провеждане на тестовите уроци в областта на вградените управляващи системи и роботиката се използва разработената в Глава 3 платформа за дистанционно обучение. Схема на постановката е представена на фигура 70.



Фигура 70 Експериментална схема

По време на експерименталните уроци в платформата за дистанционно програмиране на вградени системи и роботи се обединяват три различни локации, които улесняват дистанционното участие на всички страни.

#### 4.1 Методическа разработка и провеждане на първи урок чрез използване на платформата за отдалечено програмиране на Arduino:

##### I. Тема на урока:

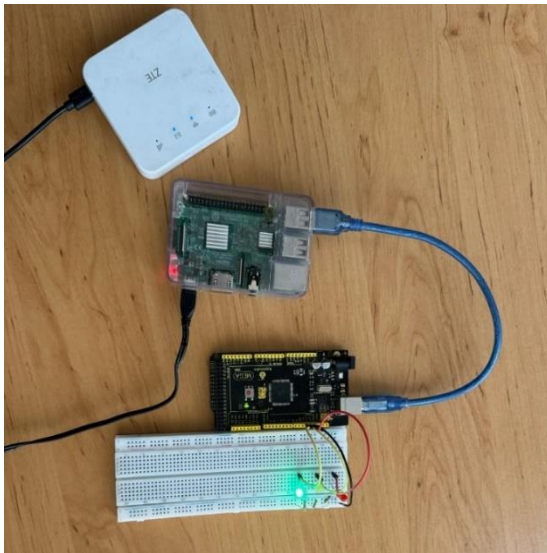
##### *Програмиране на Arduino-базирана светофарна система със светодиоди.*

На фигура 72 е показана схемата на опитната постановка. На фигура 73 и фигура 74 са представени снимки от опитната поставка върху експериментална платка.

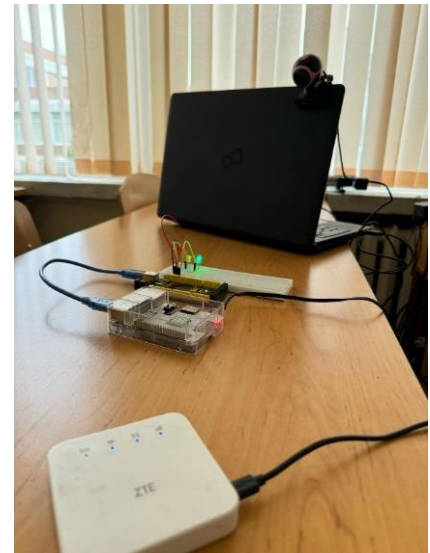
##### Цели на урока

- Разбиране на основите на управление на светодиоди:
- Развитие на алгоритмично мислене при създаването на светофарна система:
- Разбиране на серийна комуникация:

Урокът включва упражнения за мигане на светодиод, управление на светодиоди в определен ред и програмиране на основен алгоритъм за светофарна система.



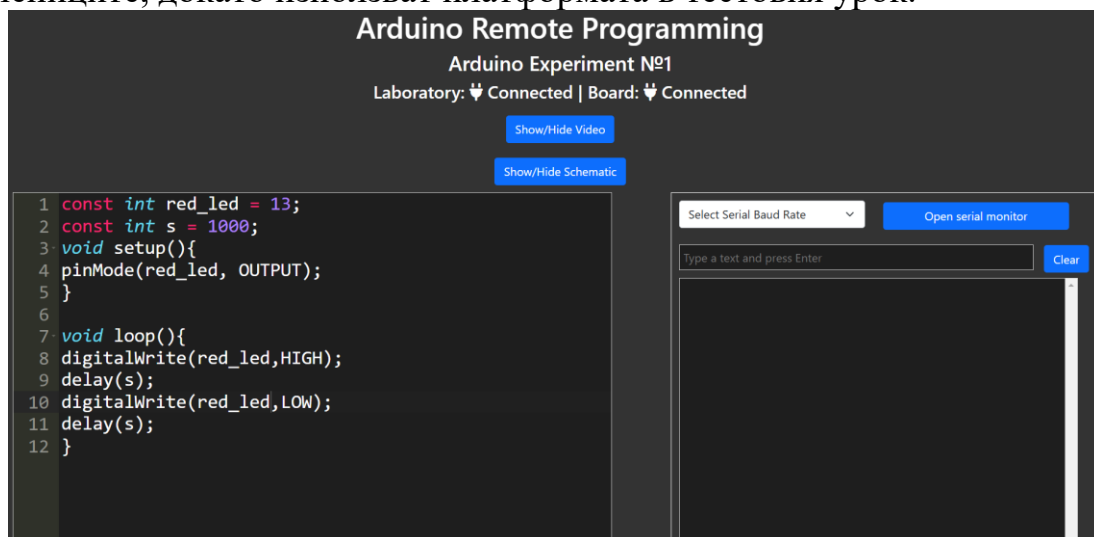
*Фигура 5 Лабораторен компютър и опитна постановка*



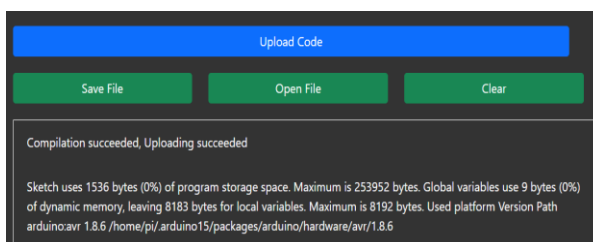
*Фигура 6 Опитна постановка и видеоизлъчване в реално време*

### Екранни снимки от провеждане на занятияето:

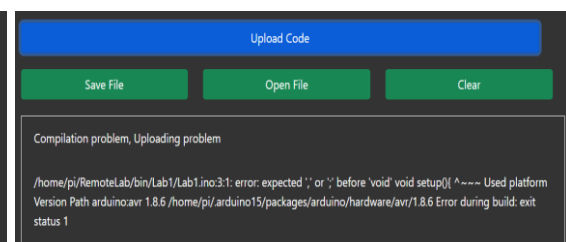
От фигура 75 до фигура 78 са представени екранни снимки от брауъра на учениците, докато използват платформата в тестовия урок.



*Фигура 7 Тестова програма за управление на светодиода*

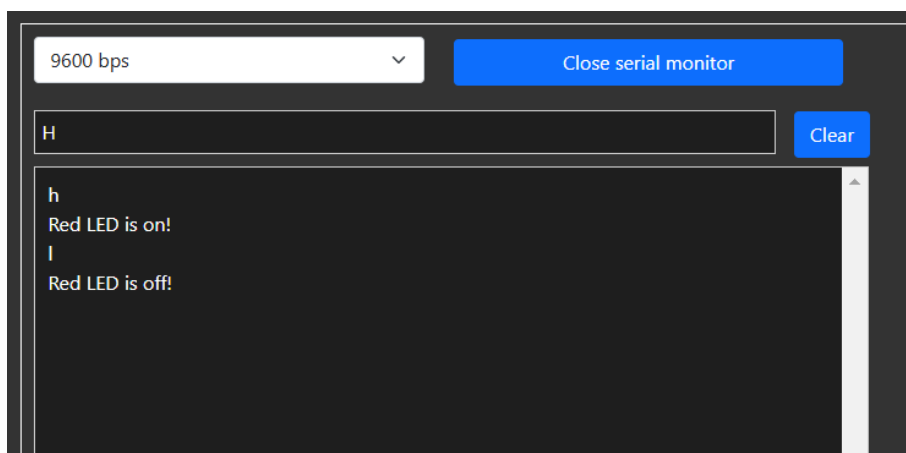


*Фигура 8 Резултат от успешно компилиране и качване на първата тестова програма*



*Фигура 9 Резултат след опит за компилация от умишлено допусната синтактична грешка - резултат*





*Фигура 78 Тестване на серийната комуникация*

В края на урока, всички тестови варианти на системата за дистанционно програмиране на Arduino бяха успешно проведени, като платформата доказва своята надеждност и ефективност в учебния процес.

## **4.2 Методическа разработка и провеждане на втори урок чрез използване на платформата за отдалечено програмиране на Arduino:**

### **I. Тема на урока:**

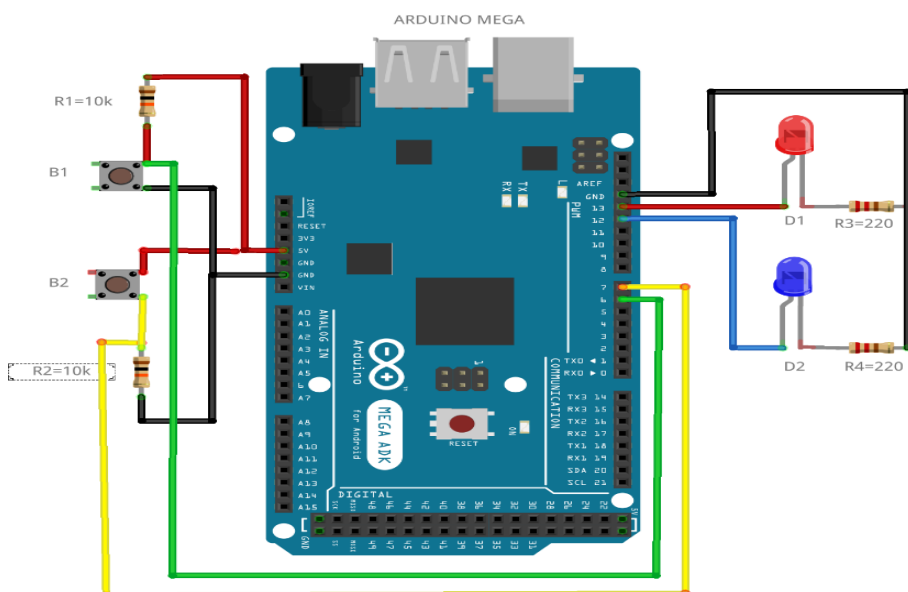
*Цифрови входове в микроконтролерите. Управление на светодиоди с бутон, използвайки Arduino и уеб-базирана система за дистанционно програмиране.*

В представения урок за лабораторно упражнение в дистанционна форма се разглежда работата с цифрови входове на микроконтролера Arduino, като специален акцент се поставя върху различните схеми на свързване на бутони с използване на pull-up и pull-down резистори.

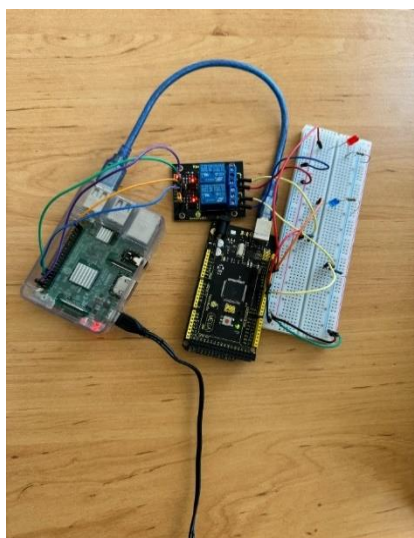
### **Цели на урока**

1. Усвояване на основите на цифрови входове в микроконтролери:
2. Изучаване на програмни конструкции за управление на светодиоди:
3. Разбиране на серийна комуникация:

На фигура 82 е показана схемата на опитната постановка. На фигура 83 и фигура 84 са представени снимки от опитната постановка върху експериментална платка.



Фигура 82 Схема на опитната постанова



Фигура 83 Опитна постанова



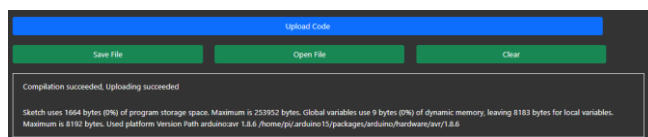
Фигура 84 Опитна постанова с видеоизлъчване

**Екранни снимки от провеждане на занятието:**

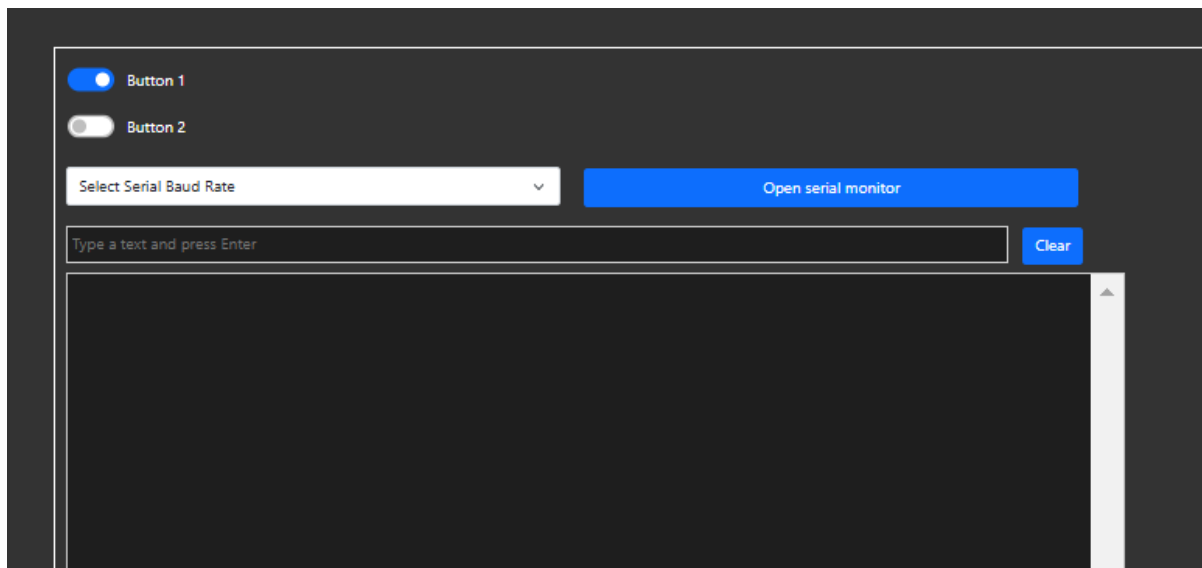
От фигура 86 до фигура 92 са представени екранни снимки от брауъра на учениците, докато използват платформата в тестовия урок.

```
1 const int buttonPin = 6;
2 const int redLedPin = 13;
3
4 void setup() {
5   pinMode(buttonPin, INPUT);
6   pinMode(redLedPin, OUTPUT);
7 }
8
9 void loop() {
10  int buttonState = digitalRead(buttonPin);
11
12  if (buttonState == LOW) {
13    digitalWrite(redLedPin, HIGH);
14  } else {
15    digitalWrite(redLedPin, LOW);
16  }
17  delay(10);
18 }
```

Фигура 86 Управление на един светодиод с бутон



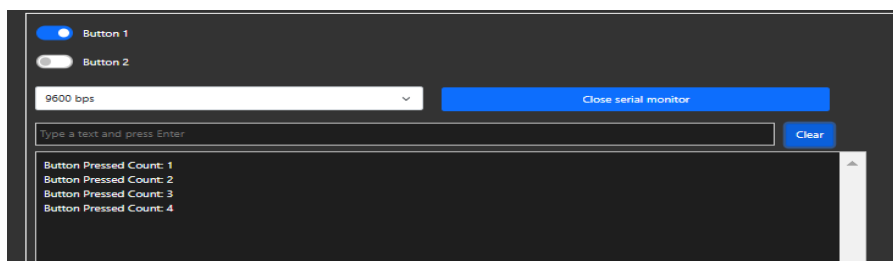
Фигура 87 Резултат от компилацията и качването



Фигура 88 Управление на емулираните бутони

```
1 const int buttonPin = 6;
2 const int redLedPin = 13;
3
4 void setup() {
5   pinMode(buttonPin, INPUT);
6   pinMode(redLedPin, OUTPUT);
7 }
8
9 void loop() {
10  int buttonState = digitalread(buttonPin);
11
12  if (buttonState == LOW) {
13    digitalWrite(redLedPin, HIGH);
14  } else {
15    digitalWrite(redLedPin, LOW);
16  }
17  delay(10);
18 }
```

Фигура 89 Умишлено допусната синтактична грешка



Фигура 92 Резултат от серийната комуникация

При провеждането на втория урок беше извършено повторно тестване на всички функционалности, наблюдавани в първия урок, като системата отново демонстрира стабилна работа при дистанционно програмиране и качване на код. Допълнително беше тествана и функционалността на системата за емуляция на бутони, които учениците могат да управляват дистанционно чрез графичния интерфейс.

#### 4.3 Методическа разработка и провеждане на урок чрез използване на платформата за отдалечено програмиране на робот Misty II:

I. Тема на урока: *“Въведение в програмирането на хуманоидния робот Misty II с Python“*

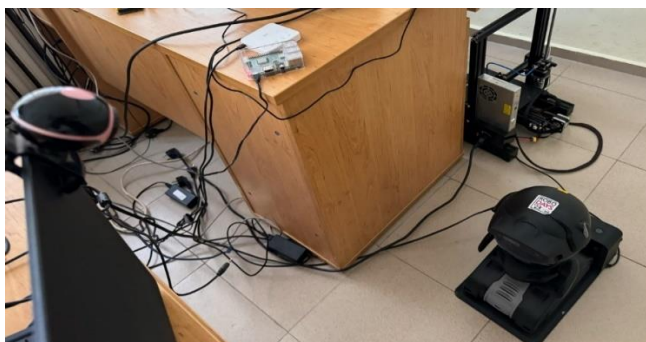
Цели на урока:

- Запознаване с основите на програмирането на хуманоидния робот Misty II чрез Python.
- Научаване на основни команди за управление и взаимодействие с Misty II, използвайки предоставената библиотека.
- Изучаване на цветовия модел RGB и управление на цветния светодиод на робота.

На фигура 93 и фигура 94 е са представени снимки от хуманоидния образователен робот, който се използва от учениците по време на тестовото занятие.



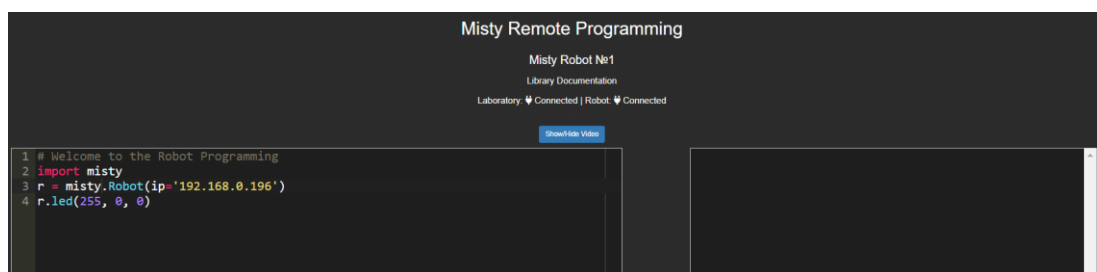
Фигура 93 Хуманоиден образователен робот Misty II



Фигура 94 Опитна постановка с Misty II

## Екранни снимки от провеждане на занятиято:

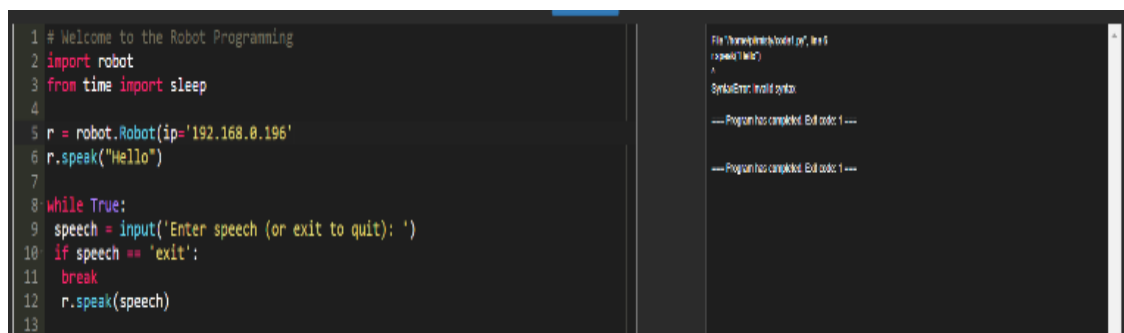
От фигура 96 до фигура 99 са представени екранни снимки от брауъра на учениците, докато използват платформата в тестовия урок.



```
Misty Remote Programming
Misty Robot №1
Library Documentation
Laboratory Connected | Robot Connected
>Show/Hide Video

1 # Welcome to the Robot Programming
2 import misty
3 r = misty.Robot(ip='192.168.0.196')
4 r.led(255, 0, 0)
```

Фигура 96 Създаване на първа програма



```
1 # Welcome to the Robot Programming
2 import robot
3 from time import sleep
4
5 r = robot.Robot(ip='192.168.0.196')
6 r.speak("Hello")
7
8 while True:
9     speech = input('Enter speech (or exit to quit): ')
10    if speech == 'exit':
11        break
12    r.speak(speech)
13

File "C:\Users\pavel\Documents\misty\misty.py", line 6
r.speak("Hello")
^
SyntaxError: invalid syntax
--- Program has completed. Edit code: 1 ---
--- Program has completed. Edit code: 1 ---
```

Фигура 99 Допусната синтактична грешка и изход от интерпретатора

## Изводи

Въз основа на проведените експерименти и разработените методически материали в четвърта глава, могат да бъдат формулирани следните изводи:

1. **Относно провеждането на експерименталните уроци:** Успешно са проведени тестови уроци в реална образователна среда с ученици от НПГ по КТС - гр. Правец;
2. **Относно техническата реализация:** Платформата осигурява стабилна работа;
3. **Относно педагогическите аспекти:** Учениците успешно усвояват практически умения чрез платформата;
4. **Относно организацията на учебния процес:** Платформата улеснява провеждането на отдалечено (дистанционно) и хибридно обучение;
5. **Относно възможностите за бъдещо развитие:** Идентифицирана е нужда от разработка на допълнителни учебни материали и осъвършенстване на платформата;
6. **Относно ограниченията и предизвикателствата:** Необходимост от стабилна интернет връзка;
  - Изисквания към техническата поддръжка на системата;

- Нужда от предварителна подготовка на преподавателите;

Тези изводи потвърждават практическата приложимост на разработената платформа и нейния принос за модернизирание на обучението по вградени системи и роботика в съвременната образователна среда.

## НАУЧНО-ПРИЛОЖНИ И ПРИЛОЖНИ ПРИНОСИ

В резултат на проведените научни изследвания в представения дисертационен труд могат да бъдат обобщени следните приноси:

- **научно-приложни приноси:**
  - предложени са архитектурни решения за разработване на нови платформи за отдалечено програмиране на вградени управляващи системи и роботи, които позволяват преодоляване на проблеми в практическото обучение по вградени управляващи системи и роботика;
  - създадени са алгоритми за основни модули за реализиране на платформи за отдалечено програмиране на вградени управляващи системи и роботи.
  - разработени са методики за хибридни уроци чрез използване на системи за отдалечено програмиране на вградени системи и роботи.
- **приложни приноси:**
  - на базата на предложените архитектурни решения е разработена тестова платформа за отдалечено обучение по вградени управляващи системи и роботика.
  - разработена е Python библиотека за програмиране на хуманоиден робот Misty II.
  - интегрирани са различни типове хардуерни устройства, включително микроконтролерни платформи (Arduino, ESP32), хуманоидни роботи (Misty II), индустриални роботи (Dobot Magician) и безпилотни летателни апарати (DJI Tello) в разработената платформа, в които са проведени експериментални изследвания в реална образователна среда за потвърждаване на ефективността на платформата.

## СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИОННИЯ ТРУД

1. Nachev V., Gotseva D. (2022), "*Design of Remote Labs and Experiments about Embedded Systems and Robotics Learning*", 10th International Scientific Conference on Computer Science (COMSCI), Sofia, Bulgaria, pp. 1-4, doi: 10.1109/COMSCI55378.2022.9912587 (индексирана в Scopus)

Цитирана в:

- Tomeo-Reyes I., Chen Z., Wijenayake C., Khatamianfar A., (2023), „*Investigating Different Approaches on Online Laboratory Practices*“, 1-8. 10.1109/TALE56641.2023.10398338.
  - Bener, R., Schmitt, P., (2024), "*Introduction of a Remote Lab for Indoor Object Position Control Based on Computer Vision Sensors and AI-Enabled Embedded Systems*", 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, pp. 1-5, doi: 10.1109/EDUCON60312.2024.10578864.
2. Nachev V. (2023), "*Web-Based Remote Laboratory for Programming Arduino-Based Experiments*", 31st National Conference with International Participation (TELECOM), Sofia, Bulgaria, 2023, pp. 1-4, doi: 10.1109/TELECOM59629.2023.10409704 (индексирана в Scopus)

Цитирана в:

- Rejón C., Martin S., Robles-Gómez A., (2024), "*Easy Development of Industry 4.0 Remote Labs*", DOI:10.3390/electronics13081508,
  - Hristov A. (2023), "*Using Python for development of an application for building and experimenting with GPSS simulation models*", 31st National Conference with International Participation (TELECOM), DOI:10.1109/TELECOM59629.2023.10409696
3. Nachev V., Gotseva D., (2023), "*Creating A Web-Based Platform For Programming Humanoid Robots*", VI International Scientific Conference on High Technology for Sustainable Development HITECH 2023, October 12-13, Sofia, Bulgaria, DOI: 10.1109/HiTech60680.2023.10759117
4. Nachev V., Gotseva D., (2024), "*Educational Remote Platform for Programming and Control of CNC 3D Printers*", TELECOM 2024 (приета за печат) (предстои да бъде индексирана в Scopus)
5. Nachev V., (2024), "*Educational Remote Platform for Programming and Control of Unmanned Aerial Vehicles* ", TELECOM 2024 (приета за печат) (предстои да бъде индексирана в Scopus)
6. Начев В., (2024), "*Лабораторни макети по електротехника, вградени системи и роботика*", Национален форум на младите педагогически специалисти „Предизвикателствата пред образованието и младите педагози в XXI-ви век”, Национална програма „Квалификация на педагогическите специалисти“ 2024 година, списание „Образование и квалификация“ 2024, с. Арбанаси, обл. Велико Търново, България – (отличена с грамота от Министъра на образованието и науката) (приета за печат)

## **SUMMARY**

### **Design and Research of Platforms for Remote Education in Embedded Control Systems and Robotics**

**Ventsislav Boykov Nachev**

The dissertation presents the research and development of a modular platform for remote programming of embedded systems and robots, intended for application in the educational process. The developed platform is based on an in-depth analysis of existing solutions and technologies, offering a flexible and scalable architecture that integrates various types of hardware devices.

The work presents the main challenges in conducting practical training in a distance learning format and outlines the advantages of remote learning platforms. Key technologies such as MQTT protocol, single-board computers (Raspberry Pi 4), and web technologies for user interface implementation are analyzed. Modules for automated programming, environment emulation, serial communication, robot control, and a heartbeat service have been developed to ensure reliable system operation.

The platform has been successfully tested in a real educational environment, demonstrating its effectiveness in supporting remote and hybrid learning, providing remote access to laboratory equipment, and enabling resource sharing among educational institutions. The identified challenges related to ensuring reliable communication, managing simultaneous user access, and guaranteeing security during remote hardware access outline the directions for future development of the platform.

The achieved results create a solid foundation for further development and application of the platform in various educational contexts, offering an integrated and scalable learning environment that supports an innovative and practice-oriented approach to teaching embedded systems and robotics.