

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ  
Фізико-математичний факультет  
Кафедра інформатики та прикладної математики

«Допущено до захисту»

Завідувач кафедри

\_\_\_\_\_ Соловйов В.М.

«\_\_\_\_\_» \_\_\_\_\_ 2020 р.

Реєстраційний № \_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_ 2020 р.

**СИСТЕМА АВТОМАТИЗОВАНОГО КОНТРОЛЮ РЕЗУЛЬТАТІВ  
ЗМАГАНЬ ЗІ СПОРТИВНОГО ОРІЄНТУВАННЯ**

Магістерська робота студента  
групи Ім-16  
ступінь вищої освіти «магістр»  
спеціальності 014 Середня освіта (Інформатика)  
**Васильєва Олега Юрійовича**

Керівник: доц., к. ф.-м.н.  
Мерзликін Павло Володимирович

Оцінка:  
Національна шкала \_\_\_\_\_  
Шкала ECTS \_\_\_ Кількість балів \_\_\_\_

Голова ЕК \_\_\_\_\_  
Члени ЕК \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ .....	3
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	6
1.1. Порядок і правила проведення змагань зі спортивного орієнтування .....	6
1.2. Аналіз наявного на ринку апаратного й програмного забезпечення для контролю результатів змагань зі спортивного орієнтування .....	9
1.3. Постановка задачі .....	13
Висновки до розділу 1 .....	14
РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ АВТОМАТИЗОВАНОГО КОНТРОЛЮ РЕЗУЛЬТАТІВ ЗМАГАНЬ ЗІ СПОРТИВНОГО ОРІЄНТУВАННЯ.....	16
2.2. Проектування апаратної складової .....	18
2.3. Проектування програмної складової.....	24
Висновки до розділу 2 .....	31
РОЗДІЛ 3. РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ .....	33
3.1. Режими роботи станцій.....	33
3.1.1. Програмна реалізація стартового та фінішного пунктів .....	33
3.1.2. Програмна реалізація контрольного пункту .....	37
3.2. Програмна реалізація заповнення турнірної таблиці .....	42
3.2.1. Встановлення та налаштування Laravel .....	42
3.2.2. Структура бази даних .....	42
3.2.3. Програмна реалізація вебсайту .....	47
3.3. Встановлення й тестування вебзастосунку .....	48
3.4. Інструкція користувача.....	49
Висновки до розділу 3 .....	51
ВИСНОВКИ .....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	54

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

Laravel — безкоштовний, PHP-фреймворк з відкритим кодом, створений Taylor Otwell і призначений для розробки вебзастосунків відповідно до шаблону model–view–controller (MVC).

Arduino — апаратна обчислювальна платформа, основними компонентами якої є плата розробки та середовище розробки.

Скетч — в середовищі Arduino програма, написана мовою C++, призначена для завантаження на плату.

## ВСТУП

Спортивним орієнтуванням називається вид спорту, який полягає в орієнтуванні на місцевості за допомогою мапи й компасу та якнайшвидшому проходженні контрольних точок, які зазвичай розташовані поблизу виразних елементів рельєфу (вершина пагорбу, поворот русла ріки тощо) й позначені помаранчево-білими мітками. Це змагання на витривалість, яке поза тим передбачає високу розумову активність (зокрема, здатність добре читати мапу та швидко ухвалювати рішення). Маркування маршруту між мітками зазвичай відсутнє, тому спортсмен прокладає свій шлях самостійно. Змагання можуть мати різну тривалість, складність рельєфу та проводитися в різний час доби чи пору року. З 1977 року спортивне орієнтування визнане Міжнародним олімпійським комітетом [1].

Незважаючи на потенціал спортивного орієнтування (можливість задіяти різні вікові категорії, відсутність потреби у спеціальній підготовці, популяризація активного відпочинку, підвищення обороноздатності держави), в Україні цьому виду спорту приділяється досить мало уваги. Зокрема, технічна підтримка змагань знаходиться на дуже низькому рівні. В багатьох офіційних змаганнях відмітка й підведення підсумків змагань здійснюються в ручному режимі, що призводить до збільшення контингенту суддів та помилок, які пов'язані з людським фактором.

Таким чином, тема роботи видається **актуальною**.

**Об'єктом дослідження** є контроль результатів змагань зі спортивного орієнтування.

**Предметом дослідження** є автоматизація контролю результатів та підведення підсумків змагань зі спортивного орієнтування.

**Мета:** створити пристрій-мітку та програмне забезпечення для контролю результатів змагань зі спортивного орієнтування.

Для досягнення мети слід розв'язати такі **задачі**:

- проаналізувати правила спортивного орієнтування та процедуру проведення змагань;
- здійснити порівняння наявних апаратних і програмних засобів для контролю результатів змагань зі спортивного орієнтування;
- сформулювати функціональні вимоги до продукту;
- обґрунтувати вибір апаратної й програмної платформ;
- спроектувати апаратну складову;
- спроектувати структури даних та алгоритми;
- створити апаратну реалізацію;
- створити програмну реалізацію;
- здійснити тестування й апробацію приладу.

**Новизна роботи** полягає в тому, що в її рамках створений прототип апаратно-програмної платформи для контролю результатів змагань зі спортивного орієнтування, яка відповідає вимогам федерації спортивного орієнтування України [3] та є унікальною в своїй ціновій категорії.

**Практичне значення** роботи полягає в тому, що створений прототип може бути взятий за основу для виробництва серійних пристроїв, що можуть бути використані під час проведення змагань зі спортивного орієнтування.

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1. Порядок і правила проведення змагань зі спортивного орієнтування

Спортивне орієнтування — складний багатокomпонентний вид спорту, в якому для досягнення високого результату необхідно в рівній мірі володіти всіма компонентами: фізичною, технічною, тактичною, психічною та інтегральною підготовленістю [2].

Основні засади організації та проведення спортивних змагань зі спортивного орієнтування на території України регулюються Правилами спортивних змагань зі спортивного орієнтування [4], створеними Федерацією спортивного орієнтування (ФСО) України. Саме цей документ був основним при вивченні предметної області.

Учасник змагань зі спортивного орієнтування з використанням спортивної мапи й компаса повинен за мінімальний час пройти заздалегідь визначену послідовність контрольних пунктів (КП) на місцевості, які складають дистанцію змагання. КП позначені на місцевості помаранчево-білими трикутними призмами з розмірами бічної грані 30x30 см. Ця мітка встановлюється на помітному на місцевості орієнтирі (вершина пагорба, яма, поворот русла ріки тощо), що позначений на мапі. Поза тим КП обладнаний засобами для відмітки про його проходження (компостери, олівці тощо). Перед початком змагань учасникові видається спортивна мапа з позначеними на ній КП. Їх кількість і послідовність проходження визначається суддями заздалегідь. Спортсмен, який проходить дистанцію з усіх передбачених КП за найменший час, визнається переможцем. Учасників зазвичай ділять на групи за віком і статтю. Наприклад, Ч40 позначає чоловіків віком від 40 до 44 років, Ж35 — жінки віком від 35 до 39 років і т.п. Повний список вікових груп і позначень наведено в [4]. Елітні групи, до яких включаються найбільш підготовлені учасники, позначаються суфіксом Е, наприклад Ж21Е — жіноча

елітна група віком від 21 до 34 років. Кожна група може бути поділена організаторами змагань на підгрупи в залежності від складності або довжини дистанції. Тоді кожна така підгрупа позначається літерою українського алфавіту, наприклад, Ж18А, Ч21Б тощо. У випадку великої кількості поданих заявок в одну групу, вона може бути поділена на паралельні підгрупи одного рівня складності, які позначають цифрами, наприклад, Ч35-1, Ч35-2 і т.д. Можуть створюватися групи для новачків, в тому числі без вікових і статевих обмежень. Вони позначаються буквою Н, наприклад, Ч16Н і т.п. Зазвичай учасник зараховується до своєї вікової групи на термін один календарний рік. У деяких видах змагань спортсмени можуть брати участь командами (наприклад, естафети).

За рівнем проведення змагання поділяються на ранги [4]:

I ранг — Ігри Олімпіад, зимові Олімпійські ігри, Параолімпійські ігри, Дефлімпійські ігри, Всесвітні ігри;

II ранг — чемпіонати світу та Європи серед спортсменів усіх вікових груп, Глобальні ігри, розіграші Кубків світу та Європи, гран-прі, Всесвітні ігри серед інвалідів, Юнацькі олімпійські ігри, Всесвітні Універсиади, Європейські спортивні ігри, інші міжнародні змагання, що включені до календарних планів міжнародних федерацій та федерацій інвалідів з відповідних нозологій;

III ранг — чемпіонати України, розіграші Кубків України, інші офіційні всеукраїнські змагання, що включені до Єдиного календарного плану фізкультурно-оздоровчих та спортивних заходів України, крім молодших вікових груп;

IV ранг — чемпіонати України, інші офіційні всеукраїнські змагання серед спортсменів молодших вікових груп, що включені до Єдиного календарного плану фізкультурно-оздоровчих та спортивних заходів України;

V ранг — чемпіонати областей, міста Київ, фізкультурно-спортивних товариств, крім молодших вікових груп, зональні змагання, що включені до відповідних календарних планів спортивних змагань;

VI ранг — чемпіонати областей, ФСТ серед спортсменів молодших вікових груп, чемпіонати міст, районів, спортивних клубів, дитячо-юнацьких спортивних шкіл, спеціалізованих навчальних закладів спортивного профілю та інші, що включені до відповідних календарних планів спортивних змагань.

На змаганнях II рангу передбачено лише використання електронних відміток (рис. 1.1), затверджених International Orienteering Federation.



Рис. 1.1. Учасниця чемпіонату світу 2018 року реєструє проходження КП за допомогою безконтактної мітки [5]



Якщо організатор має намір використовувати інший тип відміток, на це має бути отримано дозвіл спортивно-технічного комітету ФСО України.

На змаганнях III–IV рангів передбачене використання лише електронних відміток, тип яких має бути затвердженим спортивно-технічним комітетом ФСО України.

На змаганнях інших рангів дозволяється використовувати відмітки інших типів (компостери, кольорові олівці тощо).

Серед видів змагань можна виділити орієнтування бігом, орієнтування на велосипедах, орієнтування на лижах трейл-орієнтування (коли учасникам дозволено рух лише уздовж дозволених для руху шляхів або за розміткою) та рогейн (англ. rogaining — 24-годинне командне орієнтування, упродовж якого учасники мають набрати максимальну суму очок за відвідування якомога більшої кількості міток у довільному порядку). Останній може розглядатися як окремий вид спорту, позаяк його правила суттєво відрізняються від класичних правил спортивного орієнтування (довільний порядок проходження, різна кількість очок за кожен мітку, значна тривалість змагань).

## **1.2. Аналіз наявного на ринку апаратного й програмного забезпечення для контролю результатів змагань зі спортивного орієнтування**

Загальні рекомендації до систем відміток наведено в [4], хоча за згодою спортивно-технічного комітету ФСО України можуть бути використані й інші системи.

Розглянемо деякі системи електронної відмітки ідентифікації КП.

1) SFR-system orienteering (рис. 1.2) [6]. Вартість базової станції – 59 євро.



Рис. 1.2. Базова станція SFR

Компоненти системи:

Базова станція SFR U5a – простий і надійний пристрій. Станція може працювати в двох режимах:

Режим 1 – «Відмітка після натискання» – є найбільш ефективним з погляду енергоспоживання. Станція постійно знаходиться в режимі сну, натискання кнопки викликає миттєвий (час активації менше сотої частки секунди) перехід в робочий режим. Такий режим рекомендується використовувати на тренувальних заходах.

Режим 2 – «Відмітка на льоту». В цьому режимі станція постійно шукає картку. Режим рекомендований для застосування на змаганнях.

В обох режимах підтвердженням успішної позначки є звуковий і світловий сигнали.

Характеристика:

- функції: очищення, старт, фініш, контрольний пункт;
- час відмітки: менше 0,1 сек (крім очищення);
- батарея: літій-полімерна 3,7V 550 mAh;

- радіус дії: 3-10 см, залежить від типу карти;
- температура експлуатації: від -20 °С до 55 °С;
- розміри: 109 x 45 x 19 мм;
- вага: 57 g;
- ціна за станцію: 59 євро.

2) SPORTident (рис. 1.3) [7]. Вартість базової станції – 100 у.о.

Компоненти системи:

Можлива робота в безконтактному режимі SPORTident AIR + з радіусом дії до 0.5 м (при використанні чіпів SIAC).

Характеристика:

- функції: ОЧИЩЕННЯ, ПЕРЕВІРКА, СТАРТ, ФІНІШ, КП, вмикання / вимикання SIAC, ПЕРЕВІРКА БАТАРЕЇ SIAC;
- похибка таймера при нормальній температурі менше +/- 20 с на місяць;
- вбудована пам'ять на 21802 позначки, 1022 записи чіпів;
- літієва батарея, що не перезаряджається;
- термін служби батареї 3-5 років. Заміна батареї в авторизованих сервісах SPORTident;
- робоча температура -20 °С ... + 50 °С;
- клас захисту IP 64 (DIN EN 60529) – захист від пилу і захист від бризок води з усіх напрямків;
- вага 62 г;
- розмір 101 x 51 x 19 мм;
- ціна 100\$.

Система SPORTident є в списку рекомендованих ФСО України.



Рис. 1.3. Базова станція SPORTident

3) Emit (рис. 1.4) [8]. Вартість базової станції 1100NOK (понад 100 у.о.).

Базова станція: TFP (Touch Free Pro).

Характеристики:

- відстань спрацьовування: до 65 см, в залежності від швидкості спортсмена;
- живлення: чотири літієвих батареї AA, що забезпечує 5000 годин роботи;
- розмір: 220x177x22мм;
- вага: 300 г;
- ціна 1100 NOK.

Система Emit є в списку рекомендованих ФСО України.



Рис. 1.4. Базова станція TFP (Touch Free Pro).

Наявні аналоги є серійними пристроями, призначеними для підтримки змагань зі спортивного орієнтування. Вказана ціна включає в себе вартість однієї базової станції. Ідентифікація спортсмена відбувається за допомогою магнітного ключа. Основним недоліком вказаних пристроїв є їх висока ціна, що не дозволяє використовувати їх у вітчизняних змаганнях.

### **1.3. Постановка задачі**

Основною задачею магістерської роботи є розробка системи автоматизованого контролю результатів змагань зі спортивного орієнтування, яка включає в себе апаратне програмне забезпечення.

Програмне забезпечення дасть можливість покращити точність і швидкість встановлення результатів змагань.

Виходячи з аналізу наявних продуктів для контролю результатів змагань зі спортивного орієнтування, враховуючи їх переваги й недоліки, було сформульовано такі функціональні вимоги:

- реалізувати можливість реєстрації учасників змагання;
- реалізувати додаткову можливість моніторингу змагань суддівською колегією (адміністративна панель);
- реалізувати можливість запису на брелок інформації про учасника та послідовності проходження контрольних пунктів;
- реалізувати можливість підведення підсумків проведення змагань та їх оприлюднення на вебсайті;
- забезпечити конкурентну вартість створеної системи у порівнянні з наявними на ринку аналоги;
- при виборі апаратного забезпечення передбачити можливість роботи в несприятливих погодних умовах.

## **Висновки до розділу 1**

У першому розділі магістерської роботи було здійснено аналіз предметної області, який підтвердив актуальність обраної теми й дозволив зрозуміти процедуру проведення змагань зі спортивного орієнтування. Було проведено аналіз наявного апаратного й програмного забезпечення для підтримки змагань зі спортивного орієнтування, який показав, що представлені на ринку пристрої, хоча й відповідають вимогам змагань зі спортивного орієнтування, проте мають високу вартість, що робить їх недоступними для використання у вітчизняних змаганнях.

Інформаційна система, яка буде описана в подальших розділах магістерської роботи, призначена для структурного зберігання даних і

виведення інформації про результати спортсменів. Вона має усунути недоліки розглянутих аналогів, зберігши достатню функціональність і конкурентну ринкову вартість.

## РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ АВТОМАТИЗОВАНОГО КОНТРОЛЮ РЕЗУЛЬТАТІВ ЗМАГАНЬ ЗІ СПОРТИВНОГО ОРІЄНТУВАННЯ

### 2.1. Обґрунтування вибору інструментів розробки

Позаяк обробку і збереження результатів змагань планується здійснювати на віддаленому сервері, слід обрати його конфігурацію й необхідне програмне забезпечення. Враховуючи не дуже високе прогнозоване навантаження, було обрано віртуальний сервер з такими характеристиками:

- 1) ОС: Ubuntu Server 16.04 LTS 64bit;
- 2) vCPU: 1 Virtual CPU;
- 3) RAM: 1 GB;
- 4) Ємність накопичувача: 20 GB;

На сервері були встановлені пакети програм, відомі як LAMP. LAMP – акронім, що позначає набір (комплекс) серверного програмного забезпечення, широко використовуваний у Всесвітній павутині [9]. LAMP названий за першими літерами компонентів, що входять до його складу:

- 1) Linux – операційна система GNU/Linux;
- 2) Apache – веб-сервер;
- 3) MariaDB / MySQL – СУБД;
- 4) PHP – мова програмування, що використовується для створення веб-додатків (крім PHP можуть матися на увазі інші мови, такі як Perl і Python).

Акронім LAMP може використовуватися для позначення:

- 1) Інфраструктури веб-сервера.
- 2) Парадигми програмування.
- 3) Пакету програм.



Хоча спочатку ці програмні продукти не розроблялися цілеспрямовано для роботи один з одним, таке їх поєднання стало дуже популярним через свою гнучкість, стабільність, продуктивність та низьку вартість (всі складові є відкритими й можуть бути безкоштовно завантажені з Інтернету). Набір LAMP входить до складу більшості дистрибутивів GNU/Linux і надається багатьма провайдерами хостингу [10].

Система управління базою даних (СУБД) є набором програмних засобів, за допомогою якого здійснюється управління базою даних і доступ до даних.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць. До того ж у комплект поставки СУБД MySQL входить спеціальний тип таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню у СУБД MySQL постійно з'являються нові типи таблиць.

MySQL має API для мов Delphi, C, C++, Eiffel, Java, Lisp, Perl, PHP, Python, Ruby, Smalltalk і Tcl, бібліотеки для мов платформи .NET, а також забезпечує підтримку для ODBC за допомогою ODBC-драйвера MyODBC.

Для розробки бази даних в рамках магістерської роботи використовувалася СУБД MySQL 5.0.45.

У цій версії СУБД MySQL реалізована практично повна підтримка стандарту SQL [11].

На віртуальному сервері буде знаходитися база даних з результатами змагань. Внесення результатів у базу відбуватиметься з суддівського пристрою на базі мікроконтролера ESP8266. Таким чином, організаторам змагань достатньо буде забезпечити щонайменше одну точку доступу Wi-Fi у місці фінішу змагань, де учасники передаватимуть суддям свої картки після проходження дистанції.

Таким чином, при заповненні бази даних запити SQL переважно надходитимуть послідовно з одного пристрою (або кількох, якщо кількість

учасників надто велика). Очікується, що розмір бази не буде надто великим, тож потужність обраного сервера видається цілком достатньою.

## 2.2. Проєктування апаратної складової

Електронна відмітка – це пристрій читання/запису та електронний чіп для зберігання інформації. У спортивному орієнтуванні – це станції позначки й чіп, який спортсмени зазвичай надягають на палець або на руку, як браслет. Основними даними, що фіксує система, є інформація про час проходження та ідентифікатор спортсмена, які записуються в чіп.

Специфіка системи контролю результатів змагань зі спортивного орієнтування полягає в тому, що її складові мають працювати автономно (змагання можуть тривати до кількох діб) у досить несприятливому середовищі (не виключені низькі температури, висока вологість тощо). При виборі апаратних складових крім урахування суто технічних характеристик перевага надавалася недорогим стандартним компонентам, які б дозволили виробляти електронні мітки у великій кількості та здійснювати швидко заміну складових у разі виходу їх з ладу.

Для фіксації факту проходження спортсменом мітки було обрано безконтактну технологію RFID (англ. Radio Frequency Identification – радіочастотна ідентифікація) – метод автоматичної ідентифікації об'єктів, в якому за допомогою радіосигналів зчитуються або записуються дані, що зберігаються в, так званих, транспондерах, або RFID-мітках.

Така мітка, видана спортсмену, може зчитуватися пристроєм на відстані до кількох сантиметрів. Самі мітки є досить дешевими, підлягають повторному використанню й можуть бути підготовлені організаторами змагань у достатній кількості.

На етапі створення прототипу перевага надавалася готовій платі розробки. Звісно, в майбутньому, за умови серійного виробництва, може бути

розроблена спеціалізована друкована плата, але використання стандартної доступної плати має і свої переваги: при невеликих обсягах виробництва (скажімо, кілька десятків пристроїв для потреб одного спортивного клубу).

Серед найпоширеніших рішень слід відзначити Arduino-сумісні плати на базі восьмибітних мікроконтролерів сімейства AVR. Для реалізації електронної мітки для спортивного орієнтування найбільш придатними видаються плати Nano і Pro Mini та їх численні аналоги, зокрема, через компактні фізичні розміри і невисоку вартість. Остання не має роз'єму USB і потребує підключення зовнішнього програматора, що є менш зручним на етапі створення прототипу. Обидві плати найчастіше зустрічаються у версіях з мікроконтролерами ATmega328 ATmega168, які відрізняються в основному доступним обсягом пам'яті. Крім розмірів і вартості, серед переваг слід назвати відносно низьке енергоспоживання, що важливо для точки, яка працюватиме у місцевості, де відсутній доступ до джерел живлення. Суттєвим недоліком мікроконтролерів AVR є відсутність вбудованого годинника реального часу, що є вкрай важливим для спортивних змагань.

Іншим досить поширеним рішенням у тій же ціновій категорії є плати на основі 32-бітних мікроконтролерів STM32. Зокрема найбільш поширеною є плата на основі STM32F103C8T6, також відома як Blue Pill [12]. При трохи більших за Arduino Nano розмірах і майже однаковій ціні можна одержати більш потужний мікроконтролер із вбудованим годинником реального часу, хоча й дещо більшим енергоспоживанням. Серед недоліків слід відзначити те, що ці мікроконтролери, так само, як і ATmega, не мають вбудованих засобів бездротової комунікації.

Нарешті, ще одне популярне й широко поширене сімейство мікроконтролерів представлено продукцією компанії Espressif [13]. Найбільш доступними є плати на основі ESP8266 та ESP32. Останні є більш потужними й вартісними і мають додаткову функціональність (як-от вбудована підтримка Bluetooth), яку наразі не планується використовувати в проєкті. Щодо ESP8266, то він має підтримку 802.11 b/g/n Wi-Fi, що дало б

можливість організувати передачу результатів змагань до зведеної турнірної таблиці. Однак підтримка такої функціональності на кожній контрольній точці виглядає надлишковою. Тому видається доцільним використати ESP8266 лише у пристрої для суддів, за допомогою якого результати будуть зчитуватися й відправлятися на сервер.

Аналіз доступних на ринку електронних складових дозволив деталізувати архітектуру системи. Система контролю результатів змагань зі спортивного орієнтування складається з таких компонентів:

- серверна станція;
- базова станція;
- станція взаємодії з сервером;
- RFID брелки або картки.

Основні вимоги до станції взаємодії з сервером:

- світлова й звукова індикація для інформування про здійснення стирання або зчитування даних;
- передача даних з карти на сервер за допомогою Wi-Fi;
- реєстрація учасників турніру.

Принципову схему роботи станції показано на рис. 2.1.

Основні вимоги до базової станції:

- світлова та звукова індикація для інформування про проходження відмітки;
- запис номеру контрольної точки на карту;
- передача даних про місцезнаходження базової станції;
- наявність акумулятора, достатнього для безперервної роботи протягом змагань;
- здатність коректно працювати за несприятливих погодних умов (перепади температур, висока вологість тощо);
- невисока вартість складових і можливість виробництва у великій кількості.

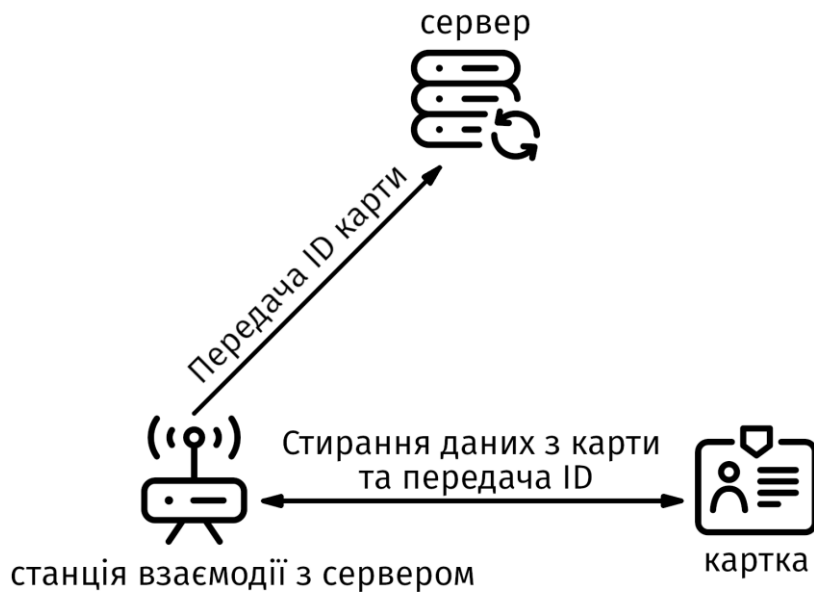


Рис. 2.1. Схема роботи приладів станції взаємодії з сервером

Конструкція базової станції:

- 1) плата з мікроконтролером stm32f103c8t6 (Рис 2.2.);
- 2) зчитувач карток RFID RC522 13,56 МГц з відстанню зчитування до 20 мм (рис. 2.3.);
- 3) світлодіод для світлової індикації;
- 4) зумер для звукової індикації;
- 5) акумулятор з платою захисту.

Конструкція станції взаємодії з ПК:

- 1) плата WeMos D1 mini Pro (Рис. 2.4.);
- 2) зчитувач карток RFID RC522 13,56 МГц (відстань зчитування до 60 мм) (Рис.2.4.);
- 3) світлодіод для світлової індикації;
- 4) зумер для звукової індикації;
- 5) інфрачервоний приймач для керування пристроєм;

б) акумулятор з платою захисту.

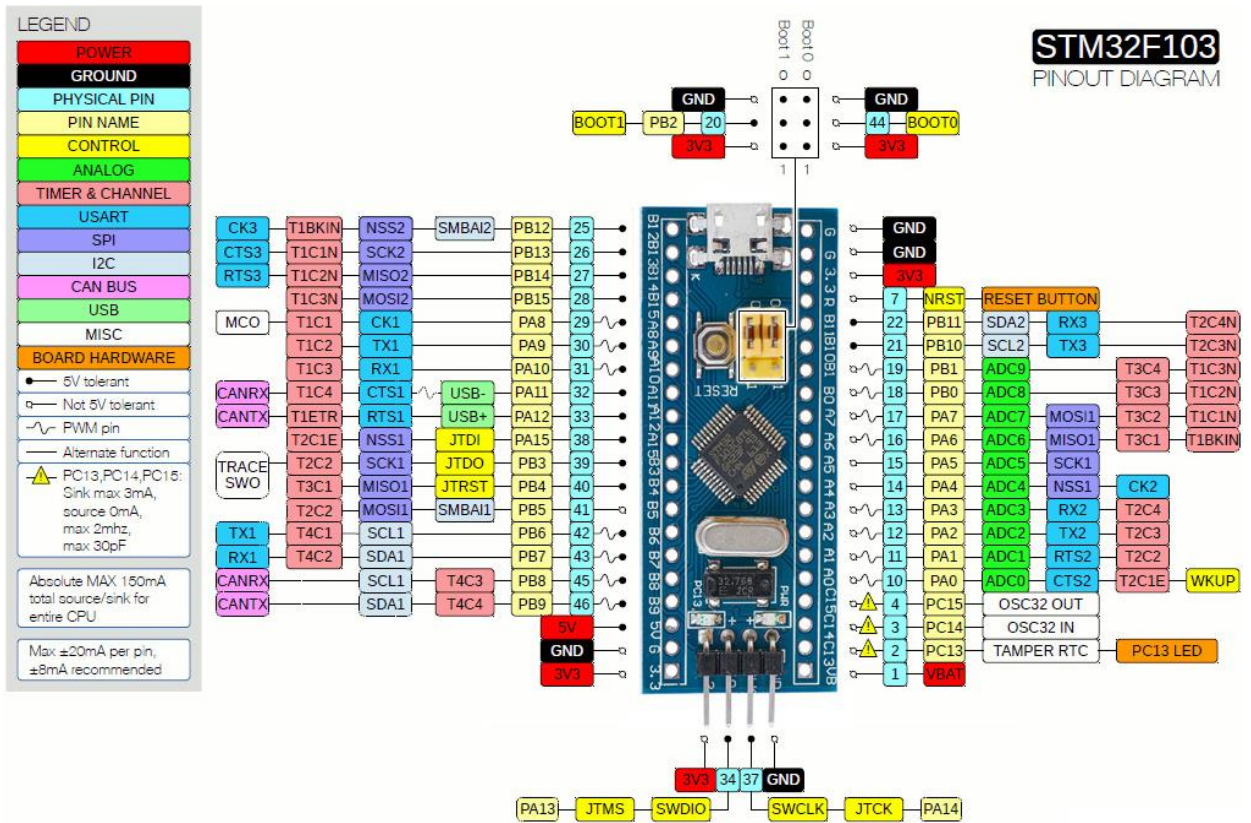


Рис. 2.2. Плата з мікроконтролером STM32F103C8T6 [14]

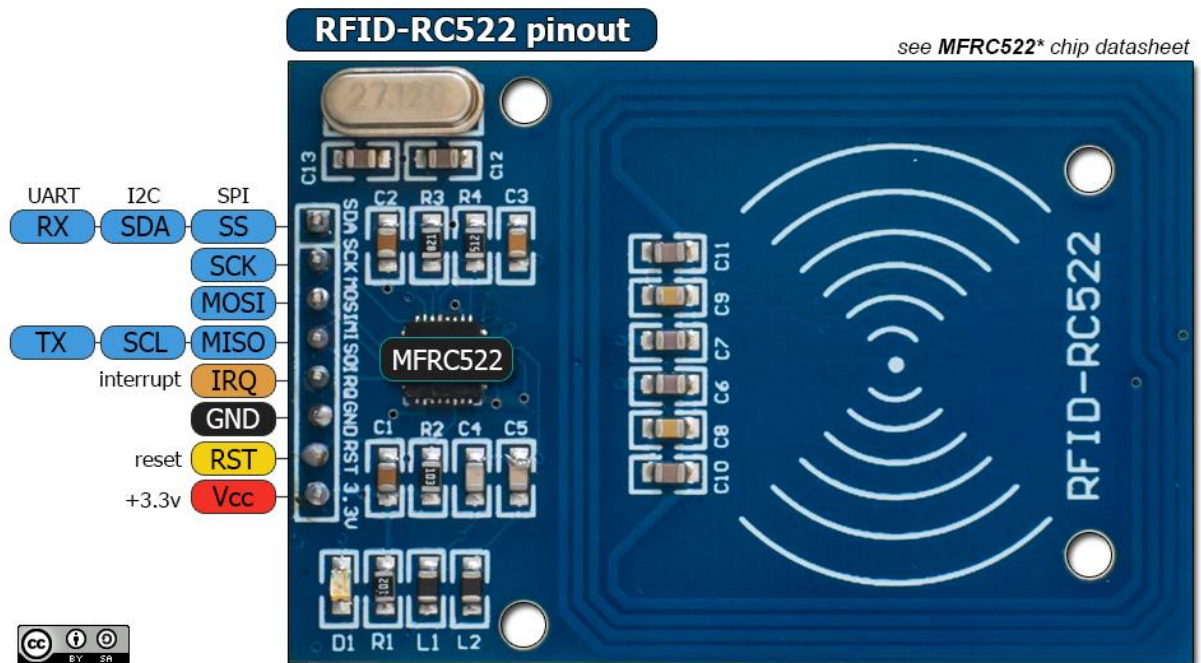


Рис.2.3. Зчитувач карток RFID RC522 13,56 МГц [15]

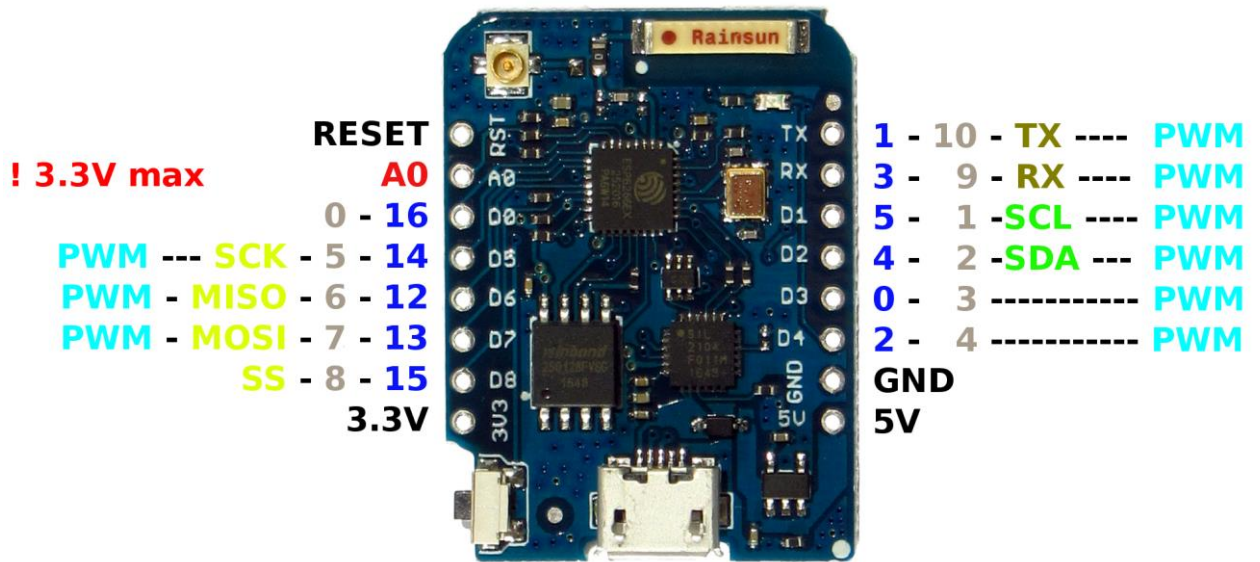


Рис.2.4. Плата WeMos D1 mini Pro [16]

Для запису даних відмітки використовуються RFID 13,56 МГц (ISO 14443A) (Рис.2.5.) картки або брелки з об'ємом пам'яті 1 Кбайт [17].



Рис.2.5. Мітка RFID 13,56 МГц (ISO 14443A) [18]

### 2.3. Проєктування програмної складової

Серверне програмне забезпечення являє собою вебсайт для реєстрації учасників та для роботи судів турніру. Для його створення було обрано мову програмування PHP.

PHP – одна з найпопулярніших у світі серверних мов програмування. Вона пройшла тривалий шлях розвитку від невеликих, вбудованих в код статичних HTML сторінок сценаріїв до сучасної мови, яка використовується в більшості сучасних динамічних вебсайтів. Складність сучасних вебсайтів постійно зростає, а функціональність розширюється, і розробляти веб-застосунки щоразу вручну, реалізуючи ті самі рутинні підзадачі, видається недоцільним. Щоб полегшити процес розробки, почали з'являтися фреймворки.

Для створення сайту було використано фреймворк Laravel. Laravel — безкоштовний PHP-фреймворк з відкритим кодом, створений Taylor Otwell і призначений для розробки веб-застосунків відповідно до шаблону model–view–controller (MVC). Серед особливостей Laravel модульна система упакування з виділеним менеджером залежностей, різні способи для доступу до реляційних баз даних, утиліти, які допомагають в розгортанні застосунків і технічному обслуговуванні [19].

MVC (Рис.2.6.) – це шаблон проєктування, в якому структура застосунку розділена на три основні компоненти:

1. Модель (model) – являє собою метод для роботи з даними.
2. Представлення (view) – користувацький інтерфейс.
3. Контролер(controller) – зв'язок між моделлю та представленням.

Laravel дозволяє з легкістю розв'язувати такі задачі:

1. Маршрутизація.
2. Обробка запитів і формування відповідей.
3. Створення представлення.
4. Доступ до різних сховищ даних.



5. Створення моделі даних.
6. Автентифікація та авторизація.
7. Зберігання та обробка даних.
8. Безпека та шифрування.
9. Обробка подій.
10. Виконання задач за розкладом.
11. Валідація.
12. Тестування.



Рис.2.6. Схема роботи фреймворку Laravel

Таким чином, фреймворк Laravel розв'язує більшість типових задач, що виникають при розробці веб-застосунку.

Поза тим, була використана система контролю версій GitHub.

Для роботи пристроїв слід написати вбудовані мікропрограми для відповідних мікроконтролерів. Для цього було обрано середовище розробки Arduino.

Arduino – торгова марка апаратно-програмних засобів для побудови простих систем автоматики й робототехніки, орієнтована на непрофесійних користувачів. Програмна частина складається з безкоштовного середовища розробки (Рис.2.7.) для написання програм, їх компіляції та програмування апаратного забезпечення. Апаратна частина являє собою набір змонтованих друкованих плат, що продаються як офіційним виробником, так і сторонніми виробниками. Повністю відкрита архітектура системи дозволяє вільно копіювати або доповнювати лінійку продукції Arduino [20].









Рис.2.7.Зовнішній вигляд Arduino IDE

Середовище розробки Arduino містить вбудований текстовий редактор програмного коду із підсвічуванням синтаксису, області повідомлень, текстової консолі, панелі інструментів для доступу до часто

використовуваних функцій і системи меню. Для завантаження програм і зв'язку середовище розробки з'єднується з апаратним забезпеченням за допомогою послідовного інтерфейсу. Зазвичай плата розробки містить чіп, що забезпечує послідовне з'єднання за допомогою більш поширеного інтерфейсу USB. Як альтернатива, може використовуватися спеціальний програматор.

Програми, створені в середовищі Arduino, часто називають скетчами. Скетч створюється й редагується за допомогою текстового редактора, що має типові інструменти для роботи з текстом (копіювання, вставка, пошук, заміна тощо). Під час збереження та експорту проєкту в області повідомлень виводяться сповіщення, попередження й помилки. Вікно консолі показує повідомлення Arduino, що включають повні звіти про помилки та іншу інформацію. Кнопки панелі інструментів дозволяють перевірити програму на наявність помилок і завантажити її в мікроконтролер, створити, відкрити і зберегти скетч, відкрити монітор послідовного порту, за допомогою якого можна з'єднуватися з платою для зневадження та обміну даними. Далі наведено основні кнопки на панелі інструментів:

-  – перевірка програмного коду на наявність помилок і подальша компіляція.
-  – перевірка програмного коду на наявність помилок, компіляція та завантаження коду в мікроконтролер.
-  – створення нового скетчу.
-  – відкриття меню доступу до всіх скетчів у блокноті. Активується натисканням у поточному вікні.
-  – збереження скетчу.
-  – відкриття монітору послідовного порту (Serial monitor).

Повний набір команд згрупований у п'ять меню: File, Edit, Sketch, Tools, Help. Доступність меню визначається станом і контекстом середовища в певний момент часу.

### **Edit**

- Copy for Discourse – копіює до буферу обміну відформатований для розміщення на форумі вихідний код скетчу з підсвічуванням синтаксису.
- Copy as HTML – копіює код скетчу до буферу обміну як код HTML для розміщення на веб-сторінках.

### **Sketch**

- Verify – перевірка скетчу на наявність помилок.
- Import Library – імпортує бібліотеку до поточного скетчу, вставляючи директиву `#include` в код скетчу.
- Show Sketch Folder – відкриває список скетчів у теці, що визначена як тека для скетчів. Її розміщення може бути змінене в налаштуваннях середовища.
- Add File... – додає файл до скетчу (файл буде скопійований з поточного місця розташування). Новий файл з'являється в новій вкладці у вікні скетчу. Файл може бути видалений з скетчу за допомогою меню закладок.

### **Tools**

- Auto Format – опція оптимізує форматування вихідного коду, наприклад, вибудовує в одну лінію по вертикалі, відкриває та закриває дужки й розміщує між ними вираз.
- Board – вибір плати, під яку буде компілюватися скетч. Користувач може встановлювати розширення для підтримки додаткових плат розробки.
- Serial Port – меню містить список послідовних пристроїв передачі даних (реальних і віртуальних) на комп'ютері. Список оновлюється автоматично щоразу при відкритті меню Tools.

- **Burn Bootloader** – пункти цього меню дозволяють записати завантажувач (Bootloader) в мікроконтролер на платформі Arduino. Ця дія не потрібна під час роботи зі стандартними платами, але стане в нагоді, якщо треба підготувати для роботи новий мікроконтролер без завантажувача. Перед записом рекомендується перевірити правильність вибору платформи з меню. При використанні AVR ISP необхідно обрати послідовний порт, до якого підключений програматор.

### **Блокнот (Sketchbook)**

Середовищем Arduino використовується принцип блокнота: стандартного місця для зберігання програм (скетчів). Скетчі з блокнота відкриваються за допомогою меню File > Sketchbook або кнопкою Open на панелі інструментів. При першому запуску середовища Arduino автоматично створюється тека для блокнота. Розташування блокнота змінюється за допомогою діалогового вікна Preferences.

### **Закладки, Файли та Компіляція**

Дозволяють працювати з декількома файлами скетчів (кожен відкривається в окремій вкладці). Файли коду можуть бути стандартними файлами Arduino, файлами C (розширення \*.c), файлами C ++ (\*.cpp) або файлами заголовків (.h).

### **Завантаження скетчу в Arduino**

Перед завантаженням скетчу потрібно обрати необхідні параметри в меню Tools > Board і Tools > Serial Port. У Mac OS X послідовний порт може позначатися як dev/tty.usbserial-1B1 (для плати USB) або /dev/tty.USA19QW1b1P1.1 (для плати послідовної шини, підключеної за допомогою адаптера Keyspan USB-to-Serial). В ОС Microsoft Windows порти можуть позначатися як COM1 або COM2 (для плати послідовної шини) або COM4, COM5, COM7 і вище (для плати USB). Додаткові параметри можуть бути налаштовані за допомогою Диспетчера пристроїв Windows. В ОС GNU/Linux порти можуть позначатися як /dev/ttyUSB0, /dev/ttyUSB1 тощо. У

разі виникнення сумнівів точну назву пристрою можна знайти в системному журналі.

Після вибору порту й плати необхідно натиснути кнопку завантаження на панелі інструментів або обрати пункт меню File > Upload to I/O Board. Сучасні платформи Arduino перезавантажуються автоматично перед завантаженням. На старих платах слід натиснути кнопку перезавантаження. На більшості плат під час процесу будуть блимати світлодіоди RX і TX. Якщо до плати підключаються периферійні послідовні пристрої, як-от модулі Bluetooth, вони мають бути від'єднані в момент завантаження мікропрограми в плату. Середовище розробки Arduino виведе повідомлення про закінчення завантаження або про помилки.

При завантаженні скетчу використовується Завантажувач (Bootloader) Arduino, невелика програма, що записується в мікроконтролер на платі. Вона дозволяє завантажувати програмний код без використання додаткових апаратних засобів. Завантажувач (Bootloader) активний протягом декількох секунд при перезавантаженні платформи і при завантаженні будь-якого зі скетчів в мікроконтролер. Про активність завантажувача можна дізнатися з блимання вбудованого світлодіоду (пін 13) (наприклад, при перезавантаженні плати).

### **Монітор послідовного порту (Serial Monitor)**

Відображає дані, що відправляються до плати Arduino та отримуються з неї за допомогою послідовного інтерфейсу. Для відправки даних необхідно ввести текст і натиснути кнопку Send або Enter. Є можливість обрати швидкість передачі зі списку, яка повинна відповідати аргументу функції Serial.begin у скетчі. На ОС Mac або GNU/Linux платформа Arduino буде перезавантажена (скетч почнеться спочатку) при активації монітору послідовного порту. Більш сучасні версії Arduino IDE підтримують також відображення даних, виводяться до послідовного порту, у вигляді графіку, що оновлюється в реальному часі.

Для написання коду програми необхідно знати, що в програмі мають бути оголошені дві обов'язкові функції: `setup()` і `loop()`.

Функція `setup()` запускається один раз, після кожного увімкнення живлення або перезавантаження плати Arduino. У тілі цієї функції пишеться код для ініціалізації змінних, установки режиму роботи цифрових і аналогових портів і т.п.

Функція `loop()` у нескінченному циклі виконує команди, які описані в її тілі. Тобто після завершення функції знову відбудеться її виклик.

## **Висновки до розділу 2**

У цьому розділі було обґрунтовано інструменти розробки, спроектовано апаратну складову та структуру системи електронної відмітки для спортивного орієнтування.

Програмне забезпечення має складатися з серверної частини і мікропрограм для плат відміток і пристрою зчитування результатів. За основу для апаратної платформи обрано плати розробки на основі 32-бітних мікроконтролерів STM32 та ESP8266 через їх достатню для реалізації проекту функціональність, доступність та поширеність, що в майбутньому дасть змогу створювати пристрої в достатній кількості за прийнятною вартістю.

Серверна складова являтиме собою вебсайт для реєстрації учасників та роботи судів турніру.

Для роботи пристроїв передбачається написання програми для мікроконтролерів, для чого була обрана платформа Arduino.

Переваги створеної системи електронної відмітки:

- невисока вартість (до 20 у.о.) порівняно з іншими системами спрощує її використання на змаганнях з низьким бюджетом або спортсменами-аматорами;

- можливість розширення й модифікації комплектації та програмного забезпечення для задоволення вимог спортсменів та суддів змагань;
- можлива модифікація комплектації для боротьби з крадіжками пристроїв на місцевості.



## **РОЗДІЛ 3. РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ**

### **3.1. Режими роботи станцій**

Для повноцінної роботи станцій було передбачено такі режими:

- 1) Для станції взаємодії з сервером:
  - a. Реєстрація RFID брелка або картки учасника змагання.
  - b. Зчитування інформації з картки та передача інформації на сервер.
- 2) Для базової станції:
  - a. Налаштування номеру контрольного пункту.
  - b. Запис номеру контрольного пункту на картку.

#### **3.1.1. Програмна реалізація стартового та фінішного пунктів**

Плати на основі ESP8266, обрані для розробки станції взаємодії, не є стандартними платами Arduino, тому перед початком програмної реалізації станції взаємодії з сервером потрібно налаштувати середовище розробки «Arduino IDE» та встановити підтримку відповідної плати. Оскільки працюємо з платою Wemos D1 mini Pro, потрібно обрати тип використовуваної плати зі списку меню «Інструменти > Плата».

У середовищі «Arduino IDE» є «Менеджер плат», який допоможе додати необхідні плати до списку. Для цього потрібно виконати такі дії:

- Відкрити «Arduino IDE» й обрати пункт меню: «Файл» > «Налаштування» (Рис. 3.1.).
- У вікні заповнити поле «Додаткові посилання для Менеджера плат» (Рис. 3.2.), ввівши в нього рядок `http://arduino.esp8266.com/stable/package_esp8266com_index.json` та натиснувши на кнопку «Ok».

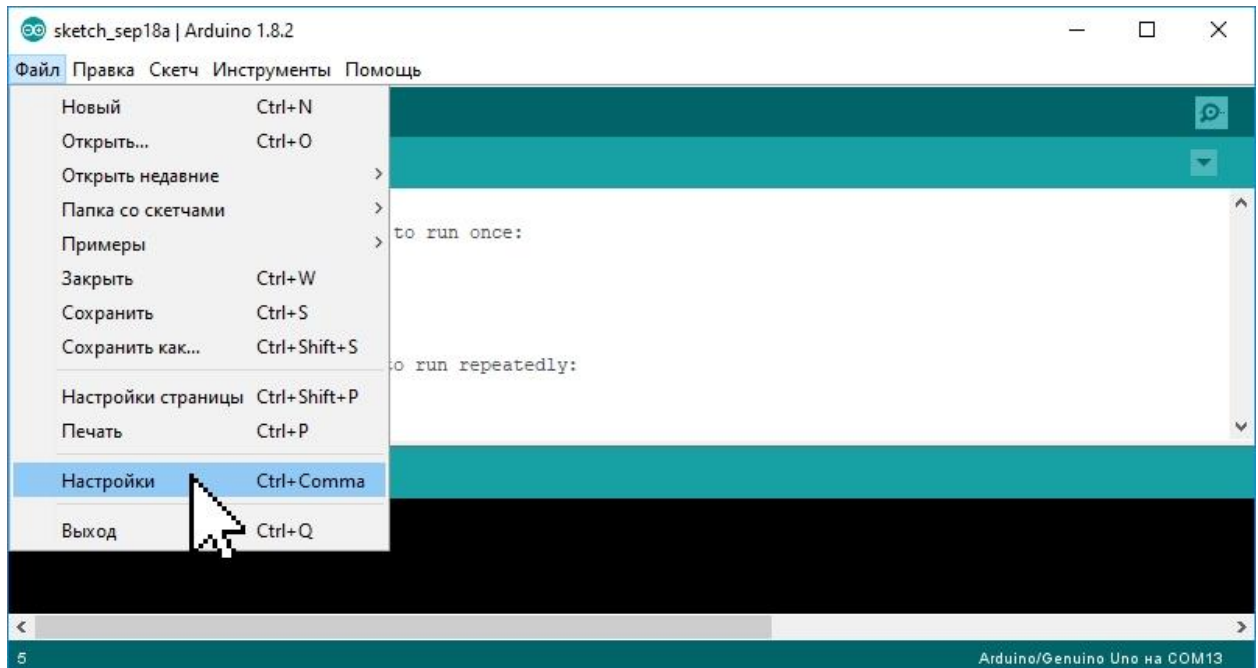


Рис.3.1. Вхід у режим налаштувань Arduino IDE

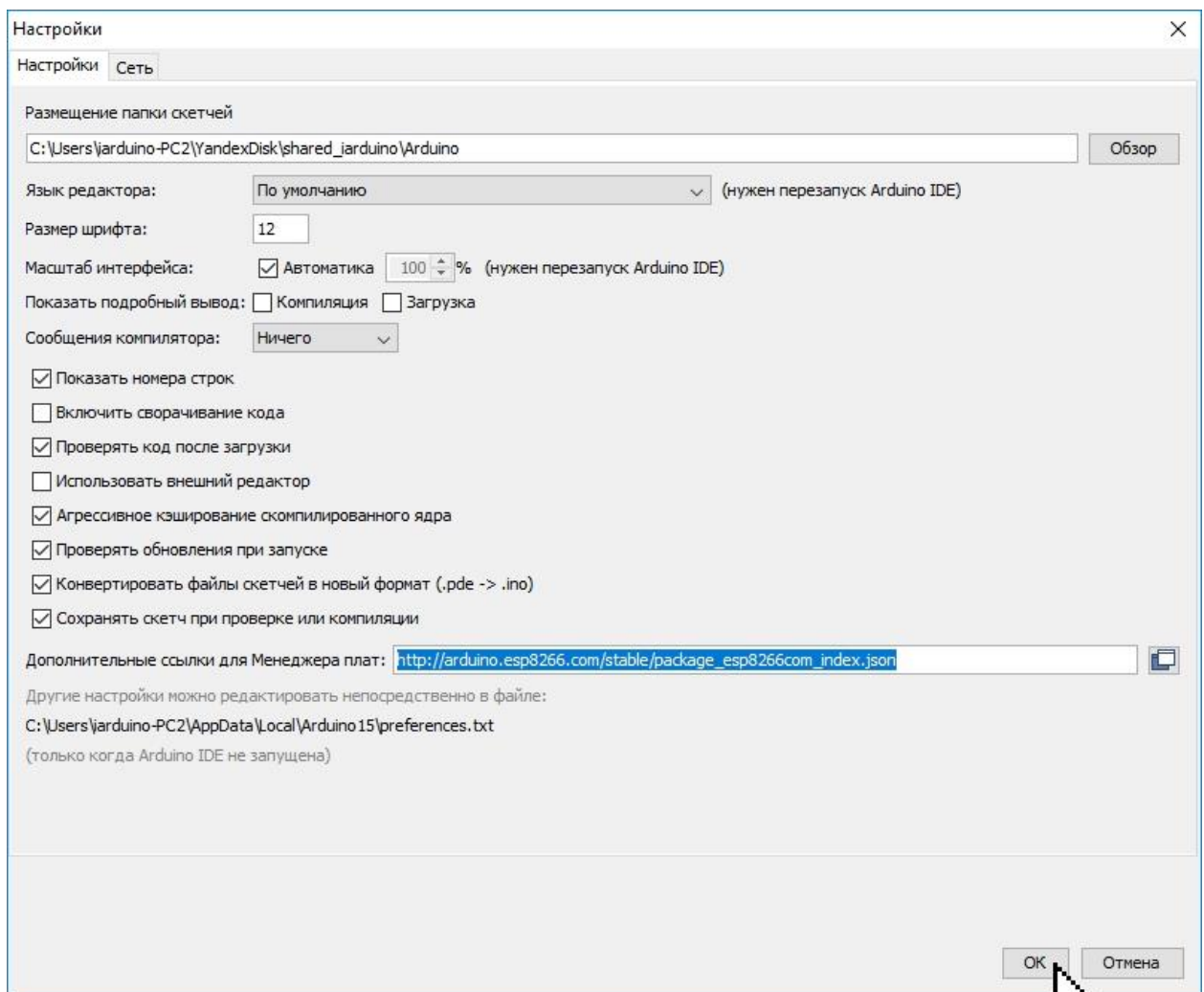


Рис.3.2. Налаштування Arduino IDE

- Запустити «Менеджер плат», обравши пункт меню «Інструменти» > «Плата» > «Менеджер плат» (Рис.3.3.).

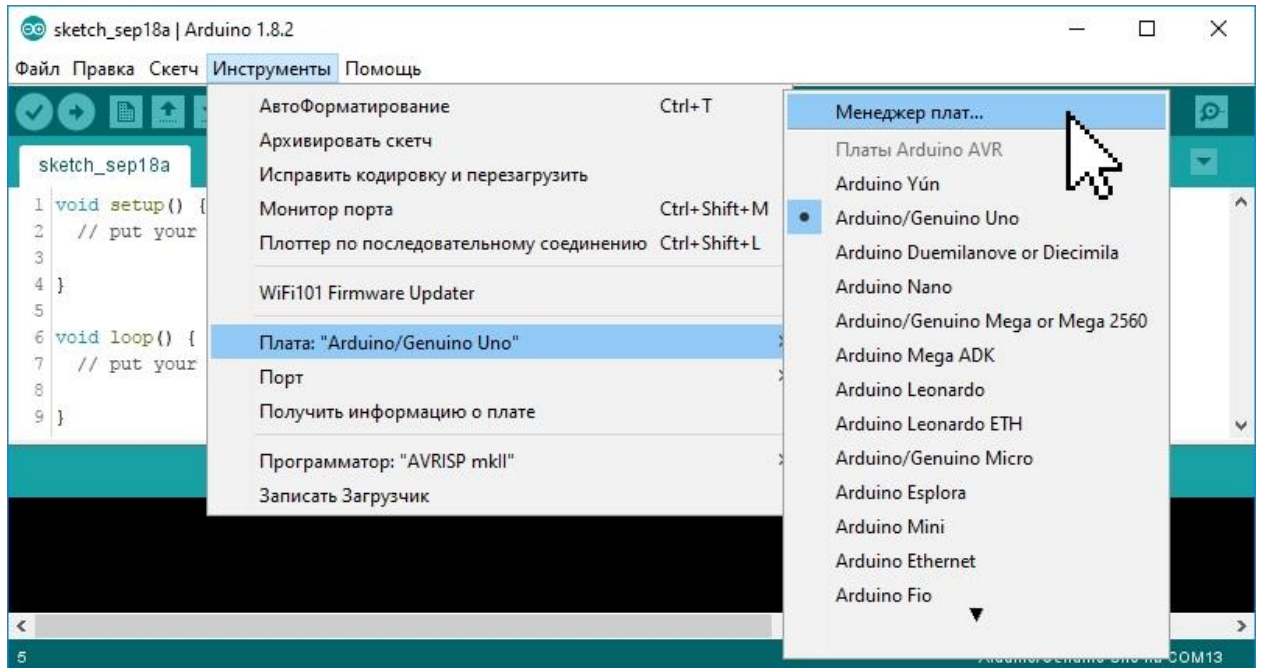


Рис.3.3. Запуск менеджера плат Arduino IDE

- У вікні «Менеджер плат» (Рис.3.4.) слід обрати «esp8266 by ESP8266 Community» зі списку й натиснути кнопку «Встановлення» (за бажання можна обрати версію встановлюваної збірки).

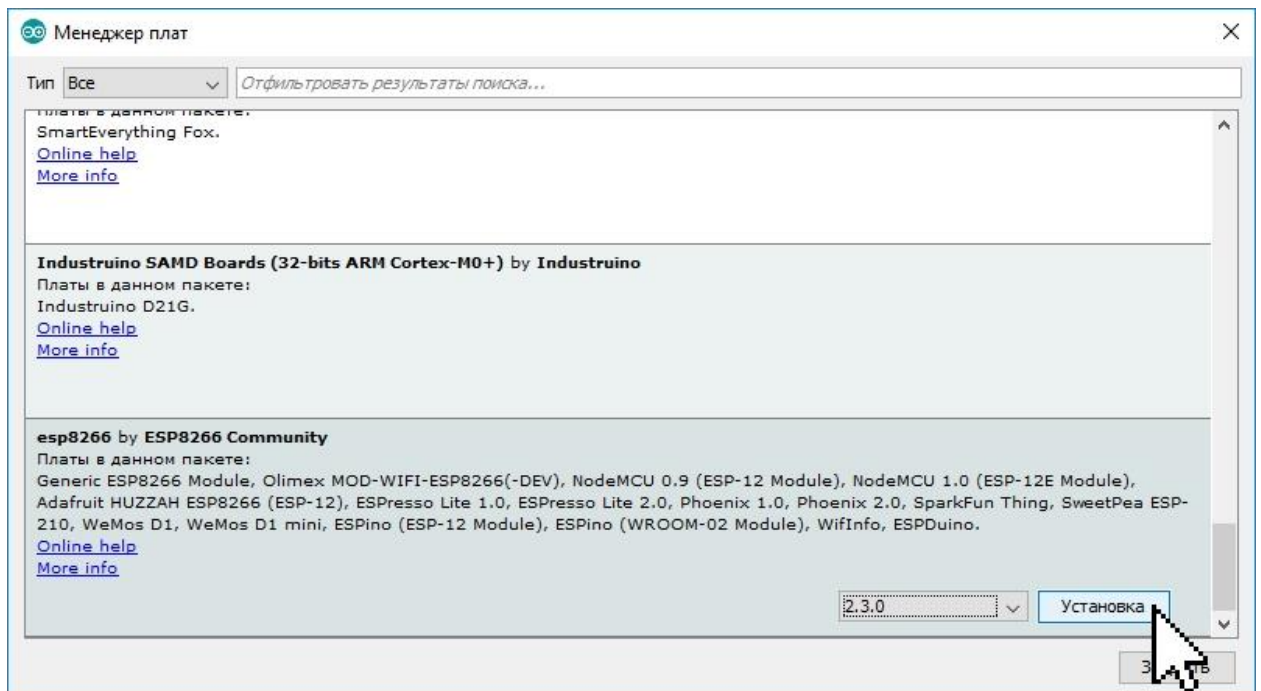


Рис.3.4. Менеджер плат Arduino IDE

- Під час встановлення «Менеджер плат» завантажить приблизно 150 Мбайт даних.

- Після успішного встановлення в списку навпроти назви «esp8266 by ESP8266 Community» з'явиться фраза «INSTALLED». Після цього слід закрити «Менеджер плат», натиснувши кнопку «Закрити» [21].

Для підтримки зовнішніх пристроїв (RFID та пульт керування) буде використано додаткові бібліотеки.

За допомогою бібліотеки MFRC522 можна зчитувати та записувати інформацію з картки.

За допомогою бібліотеки IRrecv можна керувати пультом, перемикаючи режими пристрою. Алгоритм показано на Рис. 3.5.

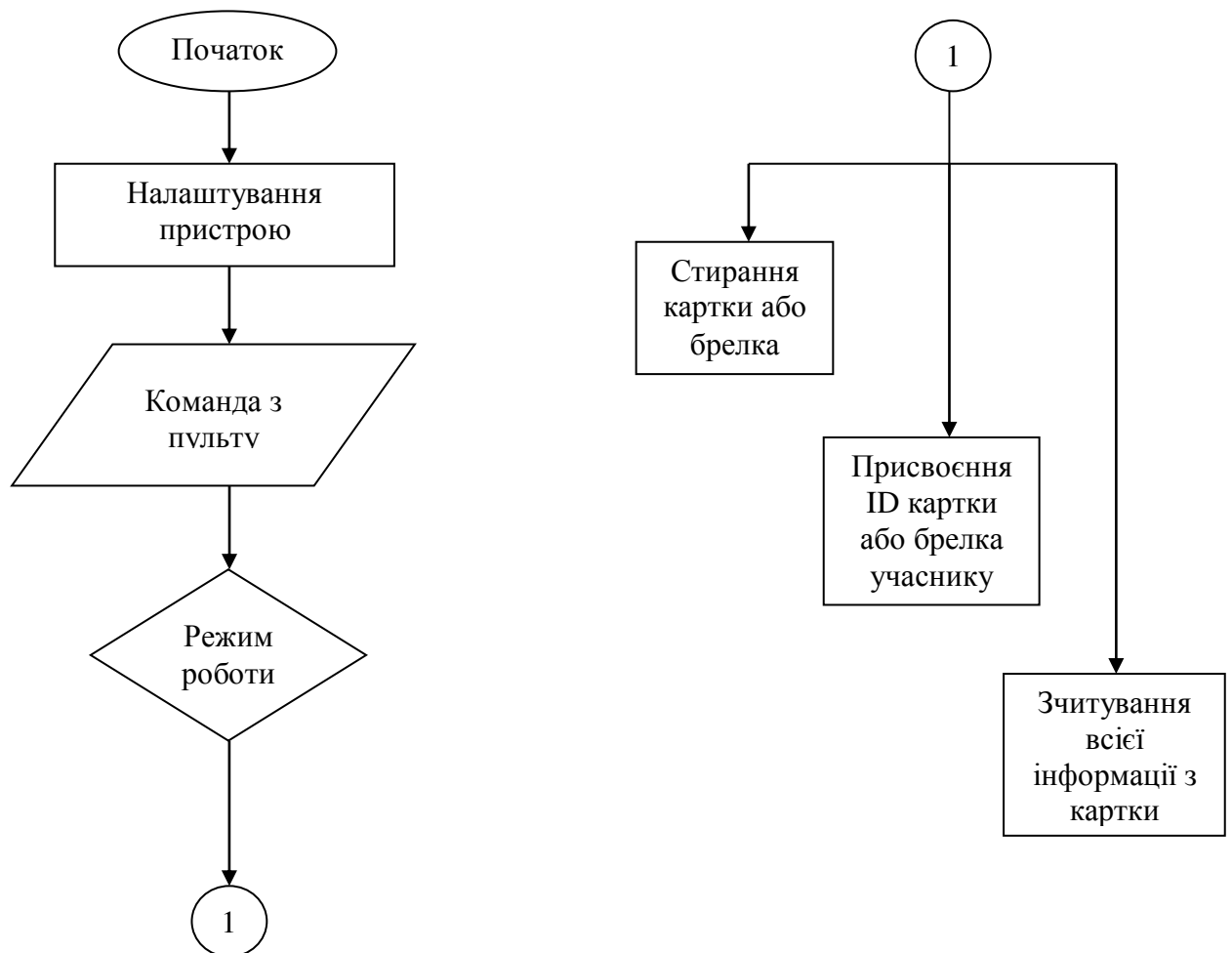


Рис.3.5. Блоксхема роботи станції взаємодії з сервером

### 3.1.2. Програмна реалізація контрольного пункту

Плата, обрана для реалізації відмітки КП, також не входить до переліку стандартних плат Arduino, тож вона потребує певної початкової підготовки для роботи з Arduino IDE. Перше, що необхідно зробити, – записати в мікроконтролер спеціальний завантажувач, який дозволить програмувати плату, використовуючи апаратний USB, причому прямо з середовища розробки. Для цього необхідно перевести перемичку «BOOT0» у положення «1», як показано на Рис.3.6.

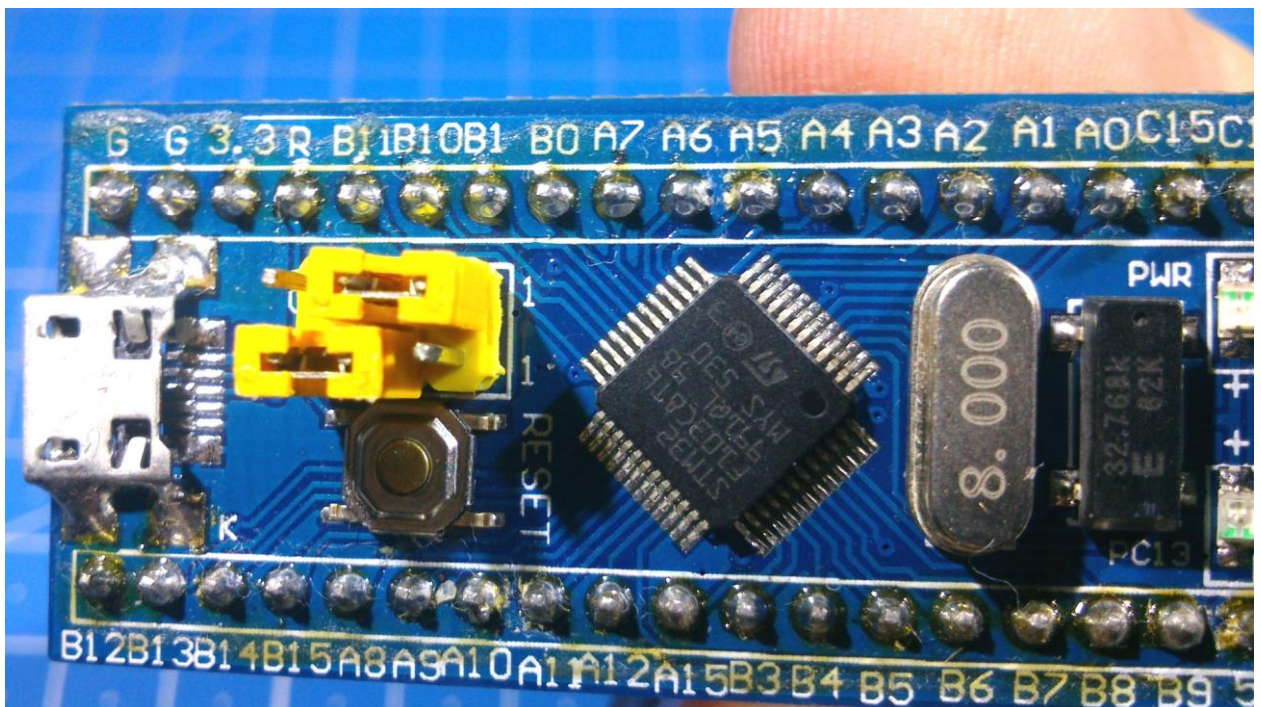


Рис.3.6. Плата STM32F103C8T6

Далі нам знадобитися перехідник з USB на UART. Варто пам'ятати, що STM32, використовує логіку 3.3 В, сумісність з п'ятивольтовою логікою не гарантована, тому рекомендовано використовувати USB to UART з можливістю вибору режимів роботи 3.3В/5В. У цьому проєкті було використано недорогий перехідник на базі чіпа CH340G.

Плату було підключено до адаптера USB to UART таким чином (Рис.3.7.):

- G <-> GND;
- 5V <-> 5V;
- PA10 <-> TXD;
- PA9 <-> RXD.

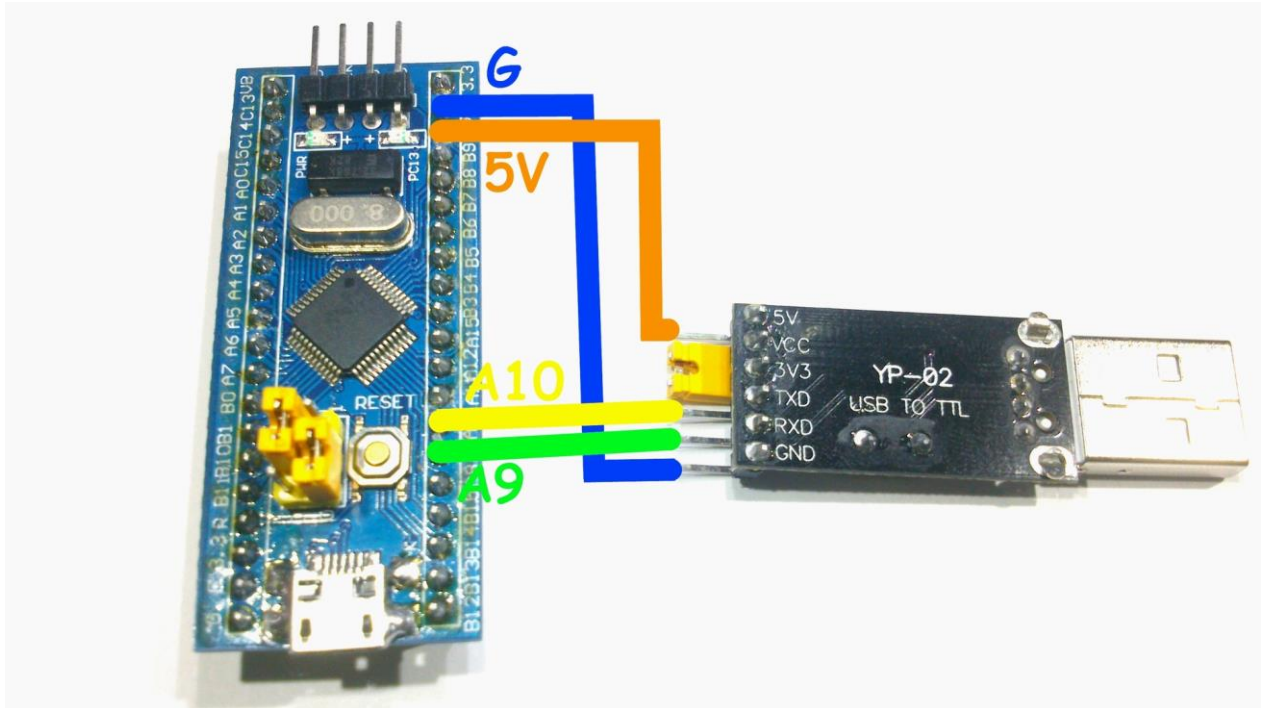


Рис.3.7.Схема підключення до USB to UART

Програма STM32 Flash Loader Demonstrator [22] (Рис 3.8) дозволяє взаємодіяти за допомогою інтерфейсу RS232 з завантажувачем, що знаходиться в системній пам'яті мікроконтролера. Для цього слід обрати номер послідовного порту, пов'язаного з нашим USB to UART адаптером, (COM43 на Рис. 3.8). Також можливо налаштувати швидкість зв'язку з портом (Baud Rate), налаштування біта парності (Parity), ехо-повторів (Echo) та інші параметри послідовного з'єднання, після чого натиснути «Next». Детальний опис протоколу USART, використаного в завантажувачі мікроконтролерів сімейства STM32, міститься в USART protocol used in the STM32 bootloader Application note (AN3155) [23].



Рис.3.8. Інтерфейс програми STM32 Flash Loader Demonstrator

Позаяк мікроконтролер новий, у нього за замовчанням встановлено захист від читання, тому програма попереджає, що після натискання кнопки «Remove protection» пам'ять Flash буде очищена, тобто у разі наявності в пам'яті якоїсь мікропрограми, вона б видалилася. У нашому випадку там нічого корисного немає, тому натискаємо кнопку (Рис.3.9.).

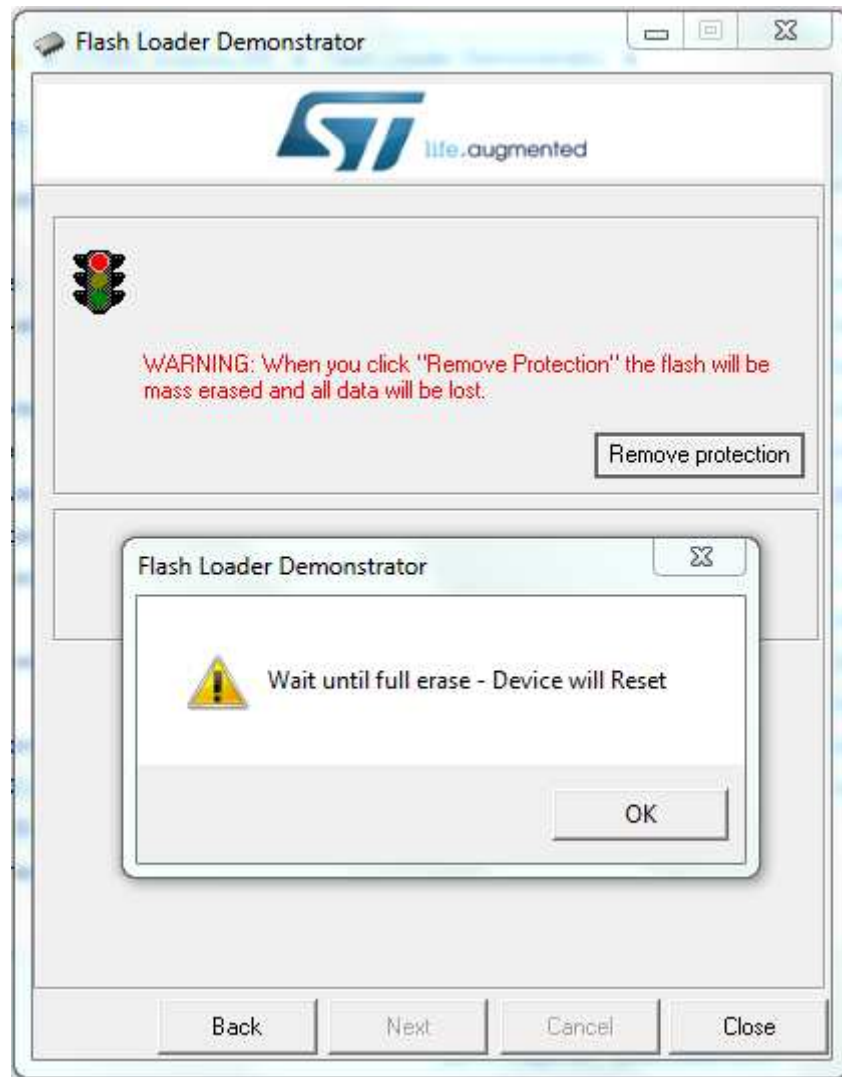


Рис.3.9. Інтерфейс програми STM32 Flash Loader Demonstrator

Після цього слід обрати мікропрограму та натиснути «Next». Потім треба завантажити спеціальне STM32 ядро для середовища розробки Arduino IDE.

Зрозуміло, що система не зможе відразу визначити пристрій, тому треба додатково встановити драйвер для плати.

Після цього перемичку «BOOT0» переводимо в положення «0» і підключаємо плату до комп'ютера за допомогою кабелю microUSB.

Запускаємо середовище розробки й обираємо пункт меню «Інструменти» > «Плата» > «Менеджер плат». Тут потрібно встановити ядро для плати Arduino Due. Обираємо останню версію й натискаємо «Встановити».



Тепер можна обрати плату зі списку: «Інструменти» > «Плата» > «Generic STM32F103C», «Variant»: «STM32F103C8 (20k RAM. 64k Flash)», Upload Method: «STM32duino bootloader» [24].

Аналогічно для розробки коду в середовищі програмування Arduino IDE потрібно підключення бібліотеки RFID та пульта керування.

Блоксхему алгоритму роботи мікропрограми базової станції показано на Рис. 3.10.

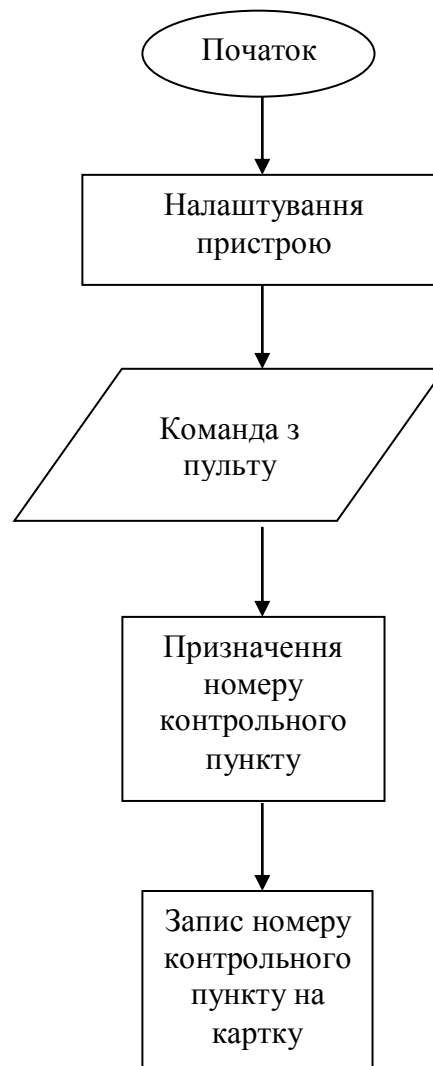


Рис.3.10. Блоксхема алгоритму роботи базової станції

## 3.2. Програмна реалізація заповнення турнірної таблиці

### 3.2.1. Встановлення та налаштування Laravel

Laravel підтримує пакетний менеджер залежностей для PHP Composer. За допомогою нього зручно керувати бібліотеками, які використовуються в проєкті, та здійснювати контроль їх версії.

У той же спосіб відбувається і встановлення Laravel. Для того, щоб завантажити встановлювач, необхідно виконати команду:

```
composer global require "laravel/installer"
```

Після чого проєкт встановлюється за допомогою команди:

```
composer create-project laravel/laravel custom
```

Після того, як Laravel успішно встановлений, слід здійснити його налаштування.

У першу чергу згенеруємо ключ застосунку. Це довільний 32-символьний рядок, який використовується в криптографічних цілях. Потім у файлі «.env» налаштуємо кешування сесій, черг; введемо налаштування для підключення до бази даних. У Laravel є вбудована команда для генерування міграції, в якій створюється таблиця зберігання даних.

Таким чином, ми налаштували середовище для розробки застосунку автоматизованого контролю результатів змагань зі спортивного орієнтування.

### 3.2.2. Структура бази даних

Вдале планування бази даних у значній мірі визначає подальшу простоту її підтримки. Дані зберігатимуться в таблицях, причому кожна

таблиця міститиме однотипну інформацію, наприклад, відомості про спортсменів.

Одним з найскладніших етапів процесу проектування бази даних є розробка таблиць, позаяк результати, які формуються в результаті запитів до бази даних (звіти, вихідні форми та ін.), не завжди можуть дати повне уявлення про те, як влаштована таблиця.

При проектуванні таблиць доцільно спланувати структуру на папері й лиш потім втілювати план у СУБД MySQL. При цьому слід керуватися такими основними принципами:

- Слід уникати дублювання даних між таблицями.

Якщо певна інформація локалізована лише в одній таблиці, то і модифікувати її доведеться тільки в одному місці. Це підвищує ефективність роботи, а крім того виключає можливість розбіжності інформації в різних таблицях. Наприклад, адреси й прізвища спортсменів мають міститися в одній таблиці.

- Кожна таблиця має містити інформацію лише про один об'єкт. Дані про кожну сутність опрацьовуються набагато ефективніше, якщо вони розташовані в незалежних одна від одної таблицях. Наприклад, результати змагань та інформація про спортсмена зберігатимуться в різних таблицях, таким чином, при видаленні результатів після закінчення змагань інформація про спортсмена лишилася в базі даних [25].

Першим і найголовнішим етапом процесу проектування та створення бази даних є розробка інфологічної моделі. Метою цієї стадії є забезпечення якомога природніших для людини способів збирання та представлення інформації, що міститиметься в майбутній базі даних. Основними структурними складовими інфологічної моделі є сутності, зв'язки між сутностями та їх атрибути.

Інфологічна модель бази даних змагань зі спортивного орієнтування показана на Рис.3.11.

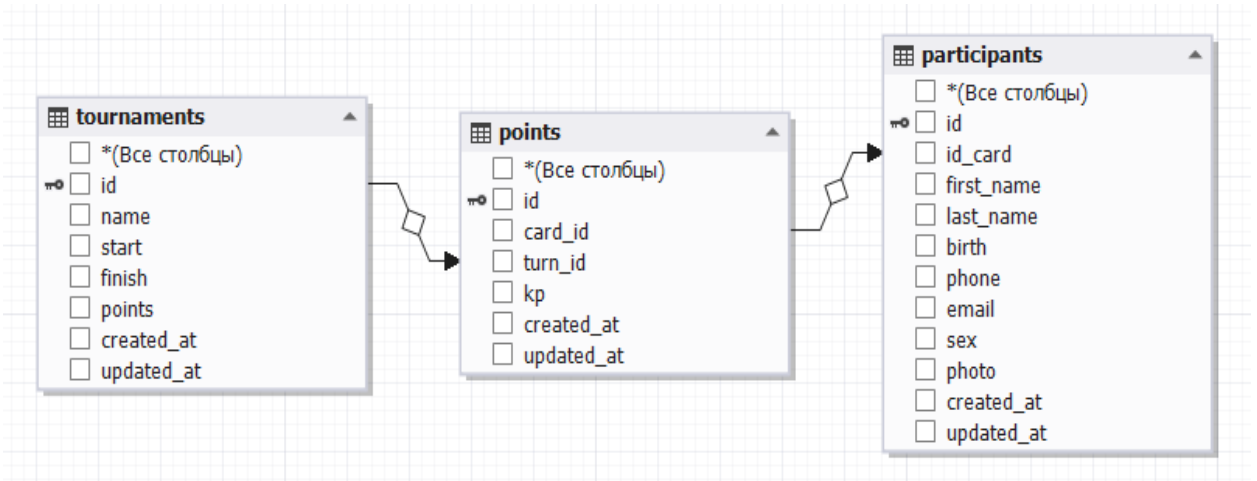


Рис. 3.11. Інфологічна модель бази даних змагань зі спортивного орієнтування

Інформація в базі даних розміщується в таблицях. Кожна таблиця складається з полів певного типу, які описують ті чи інші властивості об'єкту, якому відповідає таблиця [26]. Ключові поля призначені для однозначної ідентифікації певного запису. Таблиця «Учасники» (participants) містить інформацію про спортсменів (Таблиця 3.1).

Таблиця 3.1

#### Учасники

Назва поля	Тип поля	Розмір поля	Обов'язкове поле	Ключове поле
id	int	10	Так	Так
card_id	int	11	Ні	Ні
first_name	char	255	Так	Ні
last_name	char	255	Так	Ні
birth	date	-	Так	Ні
phone	char	50	Так	Ні
email	char	50	Так	Ні
sex	bool	1	Так	Ні
photo	char	50	Так	Ні

де, id – ідентифікатор спортсмена;  
 id\_card – ідентифікатор картки;  
 first\_name – ім'я спортсмена;  
 last\_name – прізвище спортсмена;  
 birth – дата народження спортсмена;  
 phone – телефон спортсмена;  
 email – електронна адреса спортсмена;  
 sex – стать спортсмена;  
 photo – посилання на фото спортсмена.

Таблиця «Турніри» (tournaments) містить інформацію про налаштування турнірів (Таблиця 3.2).

Таблиця 3.2

## Турніри

Назва поля	Тип поля	Розмір поля	Обов'язкове поле	Ключове поле
id	int	10	Так	Так
name	char	255	Так	Так
start	timestamp	-	Так	Ні
finish	timestamp	-	Так	Ні
point	int	10	Так	Ні

де id – ідентифікатор турніру;  
 name – назва турніру;  
 start – дата та час початку турніру;  
 finish – дата та час закінчення турніру;  
 point – кількість контрольних точок для даного турніру.

Таблиця «Результати спортсменів» (points) містить інформацію про результати спортсменів (Таблиця 3.3).

Таблиця 3.3

## Результати спортсменів

Назва поля	Тип поля	Розмір поля	Обов'язкове поле	Ключове поле
id	int	10	Так	Так
card_id	int	10	Так	Ні
turn_id	int	10	Так	Ні
kt	int	10	Так	Ні

де id\_podkat – ідентифікатор результату;

card\_id – ідентифікатор категорії;

turn\_id – ідентифікатор турніру;

kt – номер контрольної точки;

Таблиця «Судді» (users) містить інформацію про суддів (Таблиця 2.7).

Таблиця 2.7

## Судді

Назва поля	Тип поля	Розмір поля	Обов'язкове поле	Ключове поле
id	int	10	Так	Так
e-mail	varchar	255	Так	Ні
password	varchar	255	Так	Ні
fname	varchar	255	Так	Ні
lname	varchar	255	Так	Ні
phone	varchar	255	Так	Ні

де id – ідентифікатор судді;  
email – логін для входу судді;  
password – пароль для входу судді;  
fname - ім'я судді;  
lname - прізвище судді;  
phone - номер телефону судді.

### 3.2.3. Програмна реалізація вебсайту

Для зручності взаємодії з базою даних було створено вебсайт, який може переглядатися не лише з комп'ютера, але й зі смартфона. Кожен учасник змагань в режимі онлайн зможе побачити власні результати й загальні підсумки турніру.

Реєстрація учасника в турнірі показана на Рис. 3.12.

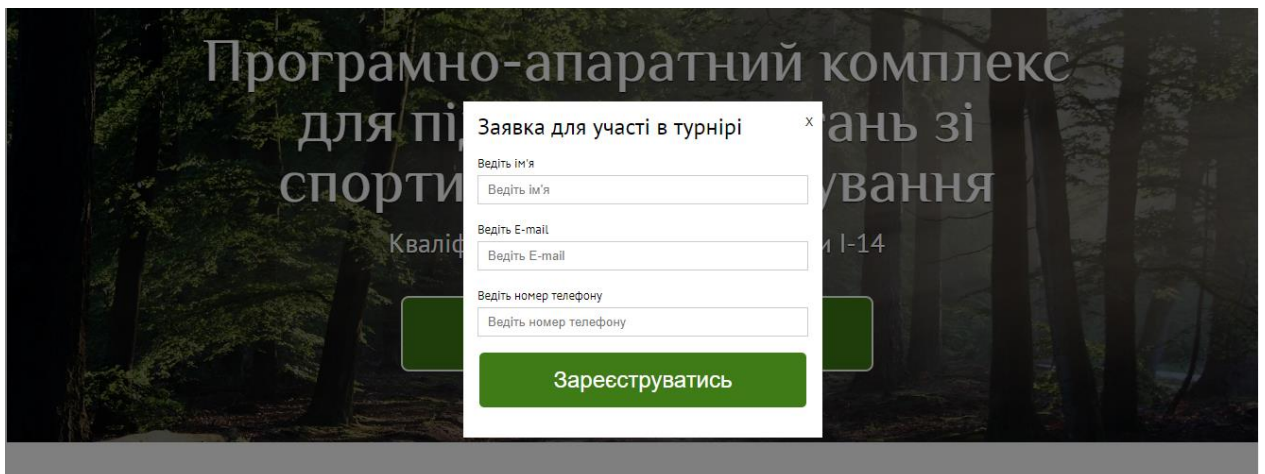


Рис. 3.12. Форма реєстрації учасника змагань

Для судів передбачена окрема адміністративна панель, яка показана на Рис.3.13.

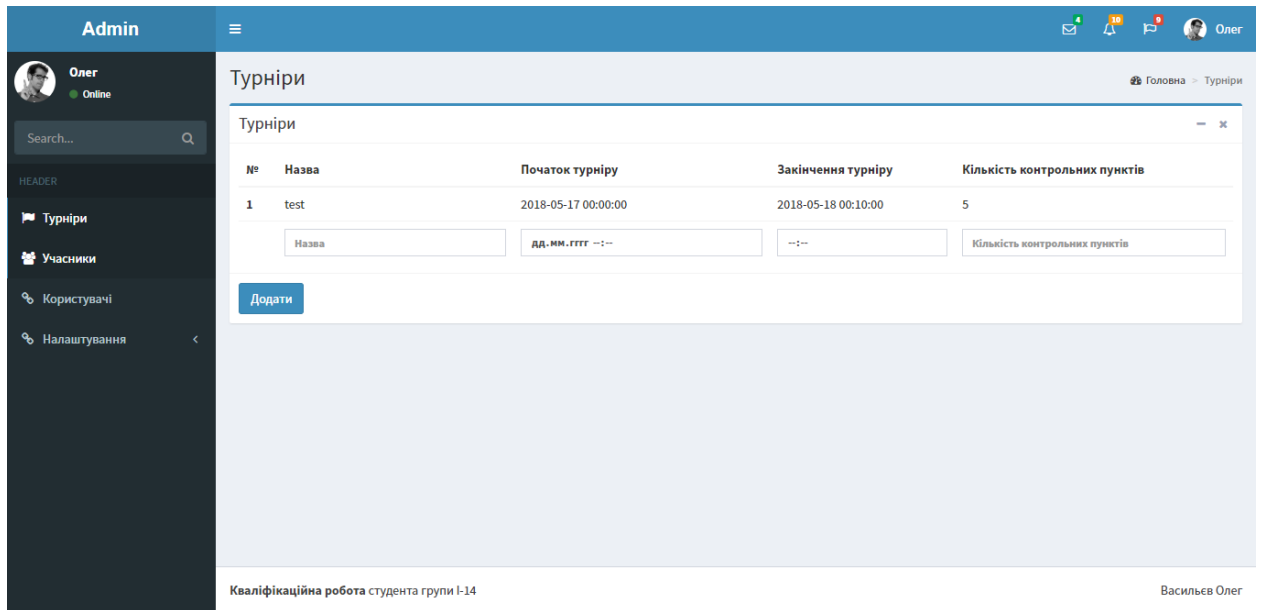


Рис.3.13. Адміністративна панель суддів змагань

Кожен розділ вебсайту найчастіше містить таблицю, в якій представлені всі записи для конкретної сутності. В інтерфейсі передбачені форми для створення й редагування записів. Для кожної таблиці доступні зручні фільтри, підтримується сортування за різними полями. Усі ці інструменти дають змогу користувачеві переглядати лише дані, які необхідні для вирішення того чи іншого завдання.

### 3.3. Встановлення й тестування вебзастосунку

#### Встановлення вебсервера Apache 2

Перше, що необхідно зробити перед розгортанням системи на вебсервері, – встановити сам вебсервер, у нашому випадку, Apache 2.

Для цього слід з'єднатися за допомогою ssh з нашим сервером та виконати такі команди:

1. Оновимо список доступних пакетів:

```
sudo apt-get update
```

2. Встановимо Apache 2:



```
sudo apt-get install apache2
```

Переконаємося, що Apache 2 встановлений: виконаємо GET запит за протоколом HTTP до нашого сервера. У відповідь отримаємо сторінку HTML від Apache 2.

### **Встановлення MySQL**

Для встановлення MySQL виконаємо команду:

```
sudo apt-get install mysql-server
```

### **Встановлення PHP**

Щойно встановлений Apache 2 не підтримує PHP процесів. Щоб виправити цю ситуацію, виконаємо команду:

```
sudo apt-get install php
```

### **Конфігурація вебсервера**

Тепер налаштуємо вебсервер, відредагувавши файл `/etc/apache2/sites-available/default`:

- указуємо файл `index.php` як індекс – у випадку запиту до публічної директорії застосунку цей файл виконуватиметься першим.
- Змінимо ім'я сервера.

Сервер налаштований.

### **Розгортання проєкту**

Весь вихідний код зберігається в репозиторії GitHub. Нам необхідно встановити git і клонувати на наш сервер.

Встановлення git:

```
apt-get install git
```

## **3.4. Інструкція користувача**

Для налаштування базової станції слід натиснути кнопку «зірочка» на пульті керування (Рис.3.14.) та номер контрольного пункту, потім натиснути

кнопку «OK». Після цієї команди номер контрольного пункту записується в пам'ять та пристрій готовий до запису карток.

Номери команд пульта керування станції з взаємодії з сервером:

1. Режим запису id-card учасника на сервер.
2. Режим прийому інформації з картки.
3. Повне стирання картки.

Для налаштування зв'язку з сервером в режимі за замовчанням плата працює як Wi-Fi маршрутизатор з вебінтерфейсом, доступ до якого можливий за SSID точки «MainStation», та паролем «admin».

Підключитися до пристрою можливо з будь-якого смартфона або комп'ютера. Далі слід у вебпереглядачі відкрити панель налаштування станції за URL <http://192.168.1.1>.

Слід обрати точку доступу до Wi-Fi, яка має зв'язок з мережею Інтернет. Після цього пристрій готовий до роботи.

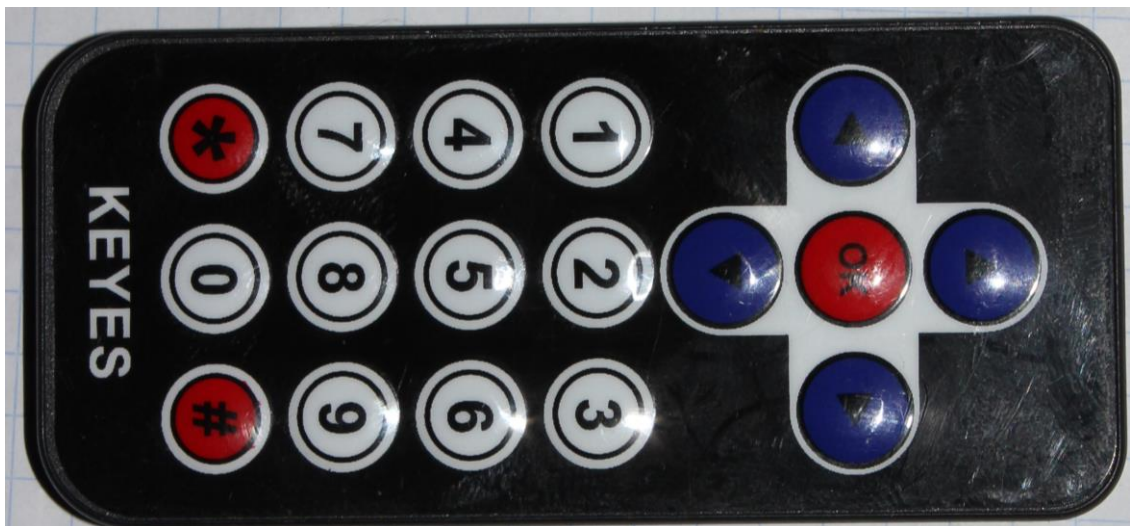


Рис.3.14. Пульт керування станціями

### **Висновки до розділу 3**

У третьому розділі описано створену інформаційну систему для автоматизованого контролю результатів змагань зі спортивного орієнтування, спроектовані таблиці бази даних, що зв'язані між собою та містять необхідну інформацію для підтримки змагань зі спортивного орієнтування.

Основними елементами бази даних є таблиці «Турніри», «Учасники» та «Контрольні точки». Ці таблиці отримують інформацію з менших таблиць, які містять інформацію про результати спортсменів.

Інформаційна система містить два види користувачів: судді та спортсмени. Адміністратор може швидко й легко додавати до бази даних нових спортсменів, нових суддів, нові турніри.

Реалізовано користувацький інтерфейс, враховуючи всі переваги та недоліки інтерфейсів вже наявних інформаційних систем. Розглянуто можливі покращення інформаційної системи в майбутньому.

## ВИСНОВКИ

Аналіз процедури проведення змагань зі спортивного орієнтування показав, що хоча на змаганнях міжнародного рівня широко використовуються подібні системи, організатори вітчизняних турнірів усе ще часто користуються неавтоматизованими КП (на основі компостерів, кольорових олівців і карток тощо).

Наявні на ринку аналоги відповідають умовам проведення міжнародних і вітчизняних змагань, але мають досить високу вартість, що робить їх недоступними під час проведення змагань національного й регіонального рівнів.

Враховуючи переваги й недоліки серійних систем, було сформульовано функціональні вимоги до власної системи для забезпечити порівняного функціоналу в більш ефективний спосіб.

Було обрано апаратну платформу на основі поширених і доступних плат розробки з мікроконтролерами сімейства STM32 та ESP8266 та безконтактної технології RFID для фіксації проходження спортсменом КП.

Для створення мікропрограм було обрано середовище Arduino через підтримку відповідних апаратних складових і наявність необхідних бібліотек.

Для зберігання результатів змагань було спроектовано базу даних і вебсайт, що забезпечує зручний доступ до неї.

Під час тестування недоліків у роботі системи не виявлено.

Таким чином, основним результатом роботи є прототип апаратно-програмної системи автоматизованого контролю результатів змагань зі спортивного орієнтування, що повністю відповідає правилам змагань і вимогам Федерації спортивного орієнтування України. Серед переваг використання створеної системи слід зазначити:

- зменшення часу відмітки спортсменами на контрольних точках у порівнянні з використанням неавтоматизованих КП;

- зменшення часу формування турнірної таблиці;
- зниження ризику фальсифікації та виникнення помилок у визначенні переможців;
- низька вартість у порівнянні з наявними на ринку аналогами;

Створена система побудована на основі стандартної елементної бази та не має аналогів у своєму ціновому діапазоні.

У подальшому є можливість модифікувати пристрої, додати радіо та GPS модулі для моніторингу базових станцій та слідкувати за учасниками в процесі змагання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Orienteering [Електронний ресурс] / International Orienteering Federation – Режим доступу: <https://orienteering.sport/orienteering/>.
2. Спортивне орієнтування [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/%D0%A1%D0%BF%D0%BE%D1%80%D1%82%D0%B8%D0%B2%D0%BD%D0%B5\\_%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F](https://uk.wikipedia.org/wiki/%D0%A1%D0%BF%D0%BE%D1%80%D1%82%D0%B8%D0%B2%D0%BD%D0%B5_%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F).
3. Федерація спортивного орієнтування України. [Електронний ресурс]. – Режим доступу: <http://www.orienteering.org.ua/>
4. Правила спортивних змагань зі спортивного орієнтування, затверджені рішенням Президії Федерації спортивного орієнтування України від 14.11.2015 / [Редакційна комісія: В. Г. Кір'янов, М. В. Опанасенко, Ю. А. Штемплер, А. Б. Калінкін, А.О. Стрижак, В. Ж. Петров, О. А. Постельняк]. – Режим доступу : [http://umansutur.at.ua/Norm-prav\\_baza/file\\_tur/pravila\\_sportivnikh\\_zmagan\\_zi\\_sportivnogo\\_orientu.pdf](http://umansutur.at.ua/Norm-prav_baza/file_tur/pravila_sportivnikh_zmagan_zi_sportivnogo_orientu.pdf).
5. World Orienteering Championships 2018 Long [Електронний ресурс]. – Режим доступу: [https://photos.google.com/share/AF1QipOCPQdg-xS-dhUVyv0TKR2bbyE7HKx1N\\_7MupBSZU\\_yeJle9yfEzpUHhyPUQPoHwA?key=c3hRbVMtbzN4YmM2N2EwS3lkUDVBSWRFY1VOWGhn](https://photos.google.com/share/AF1QipOCPQdg-xS-dhUVyv0TKR2bbyE7HKx1N_7MupBSZU_yeJle9yfEzpUHhyPUQPoHwA?key=c3hRbVMtbzN4YmM2N2EwS3lkUDVBSWRFY1VOWGhn).
6. SFR-system [Електронний ресурс]. – Режим доступу: <http://www.sportssystem.ru>.
7. SPORTident [Електронний ресурс]. – Режим доступу: <http://sportident.com>.
8. Ny TFP (Touch-Free Pro) [Електронний ресурс] / Emit – Режим доступу: <https://emit.no/nettbutikk/orientering/ny-tfp-touch-free-pro/>.
9. LAMP (software bundle) [Електронний ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/LAMP\\_\(software\\_bundle\)](https://en.wikipedia.org/wiki/LAMP_(software_bundle)).

10. Lee J. Open Source Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP / James Lee. – 1st edition. – Pearson Technology Group, 2002. – 494pp.
11. Кузнецов М. Самоучитель MySQL 5 / Кузнецов М., Симдянов И. – СПб.: «БХВ-Петербург», 2006. – 560 с.
12. Blue Pill [Электронный ресурс] / STM32-base. – Режим доступа: <https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill.html>.
13. Wi-Fi & Bluetooth MCUs and AIoT Solutions [Электронный ресурс] / Espressif Systems. – Режим доступа: <https://www.espressif.com/>.
14. Introduction to the STM32 Blue Pill (STM32duino) [Электронный ресурс] / Predictable Designs. – Режим доступа: <https://predictabledesigns.com/introduction-stm32-blue-pill-stm32duino/>.
15. RC522 RFID Reader Module [Электронный ресурс] / Microcontrollerslab. – Режим доступа: <https://microcontrollerslab.com/rc522-rfid-reader-pinout-arduino-interfacing-examples-features/>.
16. WeMos D1 mini Lite [Электронный ресурс] / arduino-projekte.info Alles über Arduino und Co. – Режим доступа: [https://arduino-projekte.info/wp-content/uploads/2017/03/wemos\\_d1\\_mini\\_pro\\_pinout.png](https://arduino-projekte.info/wp-content/uploads/2017/03/wemos_d1_mini_pro_pinout.png).
17. Васильев О. Ю. Електронна відмітка для спортивного орієнтування / Олег Юрійович Васильєв, Олександр Миколайович Степанюк // Новітні комп'ютерні технології. – Кривий Ріг : Видавничий центр ДВНЗ «Криворізький національний університет», 2017. – Том XV. – С. 254-256.
18. MFRC-522 RFID 13.56MHz čítač [Электронный ресурс] / e-radionica.com. – Режим доступа: <https://e-radionica.com/en/rfid-reader-mfrc-522-with-rfid-card.html>.
19. Laravel [Электронный ресурс]. – Режим доступа: <https://uk.wikipedia.org/wiki/Laravel>.
20. Arduino [Электронный ресурс]. – Режим доступа: <https://uk.wikipedia.org/wiki/Arduino>.

21. WEMOS Electronics [Электронный ресурс]. – Режим доступа: <https://wiki.wemos.cc>.
22. STM32 Flash loader demonstrator (UM0462) [Электронный ресурс] / STMicroelectronics. – Режим доступа: <https://www.st.com/en/development-tools/flasher-stm32.html>
23. AN3155 Application note. USART protocol used in the STM32 bootloader [Электронный ресурс] / STMicroelectronics. – Режим доступа: [www.st.com/resource/en/application\\_note/cd00264342-usart-protocol-used-in-the-stm32-bootloader-stmicroelectronics.pdf](http://www.st.com/resource/en/application_note/cd00264342-usart-protocol-used-in-the-stm32-bootloader-stmicroelectronics.pdf).
24. Arduino for STM32 [Электронный ресурс]. – Режим доступа: <http://www.stm32duino.com/>
25. Бойко В. В. Проектирование баз данных информационных систем / В. В. Бойко, В. М. Савинков. – М.: Финансы и статистика, 1989. – 351 с.
26. Тиори Т. Проектирование структур баз данных: у 2 кн. / Т. Тиори, Дж. Фрай. – М.: Мир, 1985 – . Кн. 1. – 1985.-287 с. Кн. 2, – 1985. – 320 с.