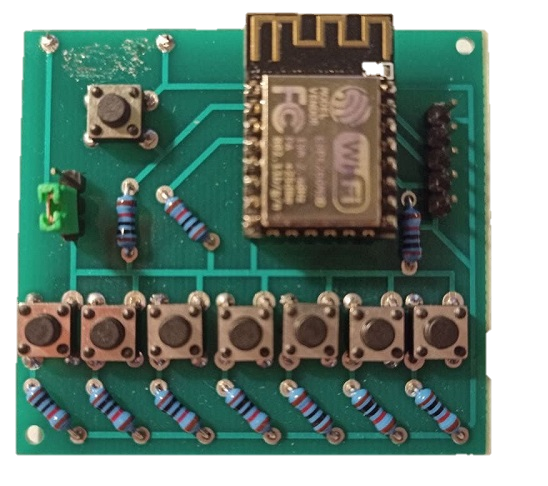
ЛІЦЕЙ №208 М. КИЄВА

**Створення пультової системи голосування**

****

**Виконав:**

Лагодич М. Д., 9-В клас

**Керівник:**

Смикало Я. Й.

Київ 2021

Зміст

[Зміст 2](#_Toc70879299)

[Вступ 3](#_Toc70879300)

[Ідея проекту 4](#_Toc70879301)

[Завдання проекту 4](#_Toc70879302)

[Хід роботи 5](#_Toc70879303)

[Створення пультів голосування та сервера 5](#_Toc70879304)

[Пульти 7](#_Toc70879305)

[Сервер 10](#_Toc70879306)

[Створення програмного забезпечення 11](#_Toc70879307)

[Для ПК 11](#_Toc70879308)

[Для сервера та пультів 13](#_Toc70879309)

[Створення засобів автоматизації 14](#_Toc70879310)

[Висновок 15](#_Toc70879311)

# Вступ

У сучасному світі є багато технологій презентування інформації. Зараз вже не є проблемою побачити на власні очі космічні станції, вимираючих рідкісних тварин та навіть ландшафти інших планет, на яких ми можемо ніколи і не побувати.

Але для якісного навчання недостатньо одного презентування. Потрібний ще і зворотній зв’язок, коли доповідач ставить запитання за щойно розказаним матеріалом. З таким підходом доповідач адаптує свою розповідь під конкретну аудиторію та навіть окремих слухачів, а аудиторія краще запам’ятовує та розуміє матеріал.

Зворотній зв’язок дуже просто отримати, коли вся аудиторія складається із 10-20 людей, тобто за допомогою піднятих рук і словесних відповідей. Та якщо доповідь ведеться на конференції на 200-300 людей, такий метод не буде ефективним. Для такого випадку зворотний зв’язок потрібно автоматизувати.

# Ідея проекту

1. Доповідач на етапі підготовки додає у свою презентацію слайди з питаннями та кількома (2-7) варіантами відповіді. Пізніше, під час доповіді, розповідач ставитиме ті запитання аудиторії. Зворотним зв’язком буде вважатися статистика вигляду «варіант – кількість людей, що обрали цей варіант».
2. Під час доповіді у кожного слухача є пристрій з кількома кнопками. Одна кнопка – один варіант відповіді. При натисканні кнопки пристрій без дротів передає обраний варіант на комп’ютер доповідача.
3. Комп’ютер доповідача малює діаграму, що буде результатом зворотного зв’язку.

Використовувати в якості таких пристроїв смартфони слухачів – не оптимальний варіант, адже не у кожного є смартфон з потрібним програмним і/або апаратним забезпеченням, яке дозволить без проблем запустити програму для зворотного зв’язку. Отже для даної задачі доцільніше створити окремі пульти з кнопками та модулями бездротового зв’язку.

Процес ззовні схожий на процес голосування, тому далі такі пульти будуть називатися «пультами голосування».

Далеко не кожен сучасний комп’ютер чи ноутбук має модуль бездротового зв’язку, тому для комп’ютера доповідача потрібно створити ще пристрій-приймач (далі – «сервер») та програмне забезпечення, що опрацювуватиме прийняті дані та виводитиме діаграму.

# Завдання проекту

Створити пульти голосування, сервер, програмне забезпечення для них та для комп’ютера доповідача. Також створити документацію проекту та автоматизовані засоби для простого повторення проекту іншими людьми.

# Хід роботи

## Створення пультів голосування та сервера

Пульти та сервер повинні мати бездротовий зв’язок, при чому такий, який дасть змогу підключити десяти пристроїв та передавати інформацію без втрат, а також бажано, щоб кожному пристрою-клієнту присвоювався певний ідентифікатор, за допомогою якого система зможе записувати відповіді окремих респондентів.

На сьогоднішній день дуже розповсюджений протокол Wi-Fi. Він цілковито підходить під мої потреби, а до того ж систему з Wi-Fi можна буде відлагоджувати за допомогою звичайного смартфона, адже багато сучасних телефонів сумісні з ним.

Отже кожен пульт має складатися із блоку керування та WiFi-модуля. Блок керування слідкуватиме за натисканням кнопок і транслюватиме номер натиснутої кнопки серверу через WiFi-модуль.

Такий алгоритм виходить занадто складний для виконання простою електричною схемою, тому блоком керування має виступити мікроконтролер – програмуєма мікросхема, що є повноцінним комп’ютером.

Існує мікроконтролер під назвою ESP8266, що є програмуємим модулем Wi-Fi, тобто один пристрій виконує одразу дві функції. Модуль достатньо відомий, для нього існує зручне середовище програмування та прості у використанні бібліотеки. Отже мій вибір упав на модуль ESP8266.

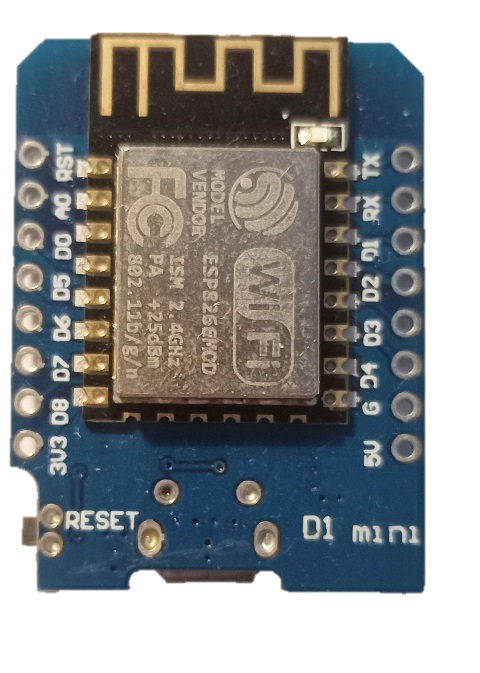


Рисунок . Модуль ESP8266 на платі-перехіднику

Слід зазначити, що такий модуль може виступати і в ролі керуючої частини пульта, і в ролі сервера. Будучи в ролі сервера, ESP8266 може створити власну WiFi-мережу (в режимі так званої *точки доступу*) або користуватися сторонньою мережою (в режимі *клієнта*), наприклад впровадженої звичайним WiFi-роутером. Можливість переходити в різні режими слід буде врахувати при написанні програмного забезпечення.

Стандартні бібліотеки для даного модуля реалізують лише один Інтернет-протокол – HTTP. Отож тільки за допомогою моделі «сервер-клієнт» комунікуватимуть сервер та пульти.

Сервер потребуватиме під’єднання до комп’ютера доповідача. Сьогодні де-факто стандартом комунікування периферичних пристроїв з персональними комп’ютерами є протокол USB, тому саме його матиме використовувати сервер.

Проте сам по собі модуль ESP8266 не може з’єднуватись по USB, бо потребує іншогої напруги живлення (3,3 В, а для USB застосовується 5 В) та розрахований на інший протокол обміну даними (UART). Для вирішення даної проблеми існують плати-перехідники (*рис. 1*), на які WiFi-модулі впаюються початково, на заводі. Такий комплекс вже повністю готовий до програмування та подальшої роботи. Тільки для роботи з ним на комп’ютері має бути встановлений спеціальний драйвер.

Отже, тепер частина системи, периферична до комп’ютера доповідача, визначена.

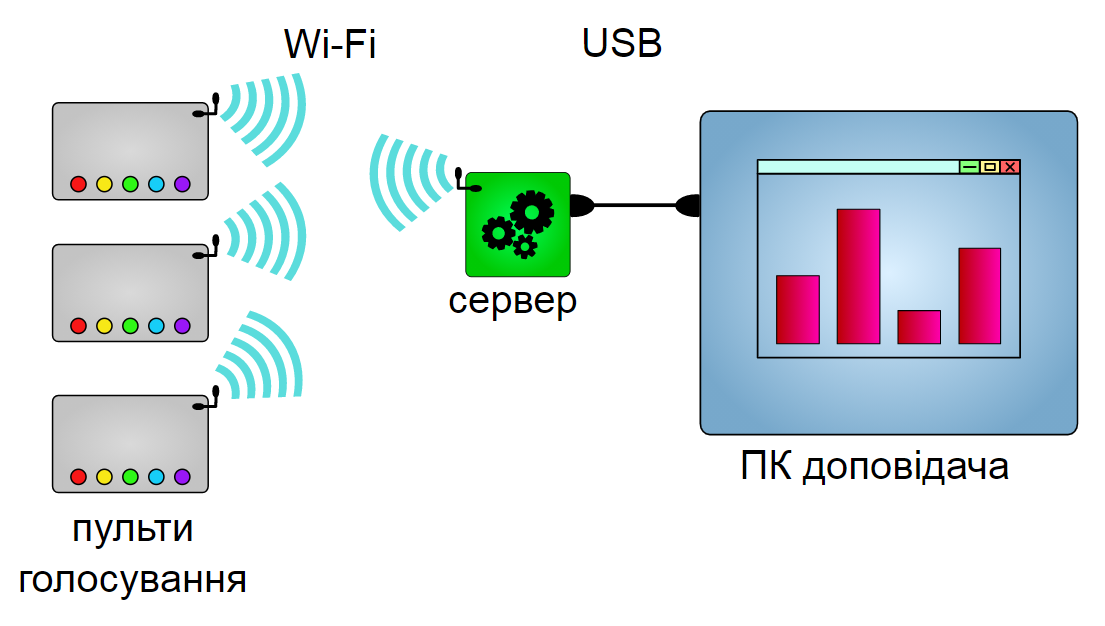


Рисунок . Загальна структура системи

### Пульти

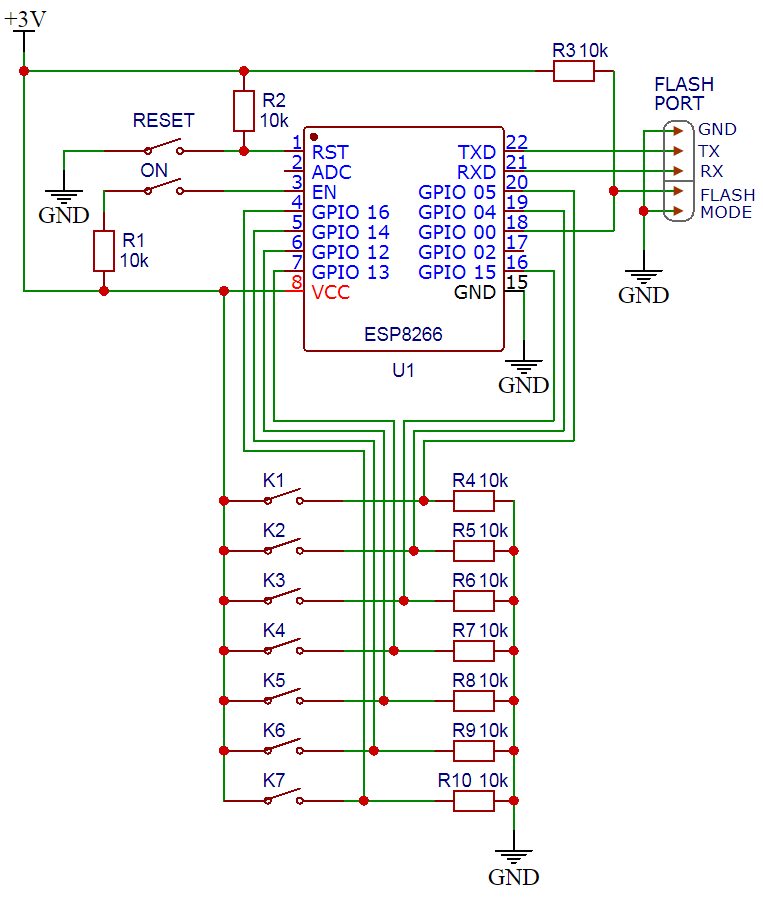


Рисунок . Схема пульта

Внутрішню будову пульта поясню за принциповою схемою (*рис. 3*).

*Живлення* до модуля ESP8266 (U1) під’єднується до контактів VCC та GND, але модуль для увімкнення потребує подачі високого рівня сигналу на контакт EN (від англ. Enable - увімкнути).

*Перезавантаження* необхідне для пульта, як і для будь-якого пристрою. Роль перезавантажуючого механізму виконує кнопка RESET, що при замиканні подає низький рівень на контакт RST (від англ. Reset - перезавантажити). Зауважимо, що для коректного спрацьовування необхідне підтягнення контакту RST до високого рівня резистором R2 (детальніше про підтягнення див. нижче).

*Перезапис програмного забезпечення* реалізується за допомогою порта FLASH (від англ. спалах, тут: швидкий запис). Контакти GND, TX та RX під’єднуються за допомогою UART-USB перехідника до комп’ютера із ПЗ, потрібним для перезапису. Контакти, об’єднані під назвою FLASH MODE, замикаються для введення модуля ESP8266 в режим програмування.

*Клавіатура* із кнопок К1-К7 слугує для введення відповіді. При натисканні кожна кнопка подає сигнал високого рівня на відповідний контакт GPIO (від англ. General Purpose Input/Output – загальний ввід/вивід). Для введення не використовуються контакти GPIO 00 та GPIO 02, бо вони є службовими (GPIO 00 при ініціалізації ESP8266 вводить його в режим програмування, якщо містить низький рівень, а до GPIO 02 під’єднаний світлодіод на корпусі, і його підтягнення викликає помилку ініціалізації).

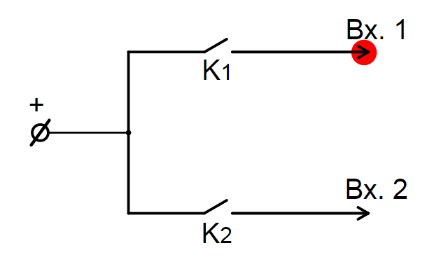
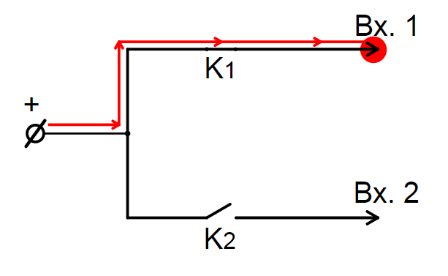
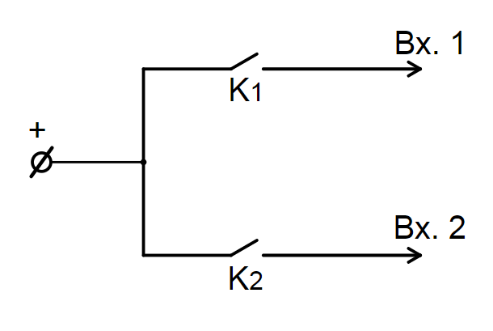
*Підтягнення* – це механізм втримання на контакті рівня сигналу, що є протилежним до рівня спрацьовування. Розберімо проблему на прикладі кнопок:

*(в)*

*(б)*

*(а)*

Рисунок . Поведінка пристрою без підтягування



Нехай існує кнопка К1 (*рис. 4.а*), що при замиканні подає високий рівень на вхід 1 (*рис. 4.б*). При розімкненні кнопки (*рис. 4.в*) на вході 1 лишається заряд, що сприймається мікроконтролером як сигнал високого рівня. Таким чином, кнопка може спрацювати лише один раз, що є проблемою.

Приєнаймо вхід 1 до негативного полюса джерела живлення через резистор R1 (такий резистор називається *підтягуючим*). Після розімкнення кнопки К1 залишковий заряд стіче до мінуса, і мікроконтролер сприйме відсутність заряду на вході 1 як низький рівень.

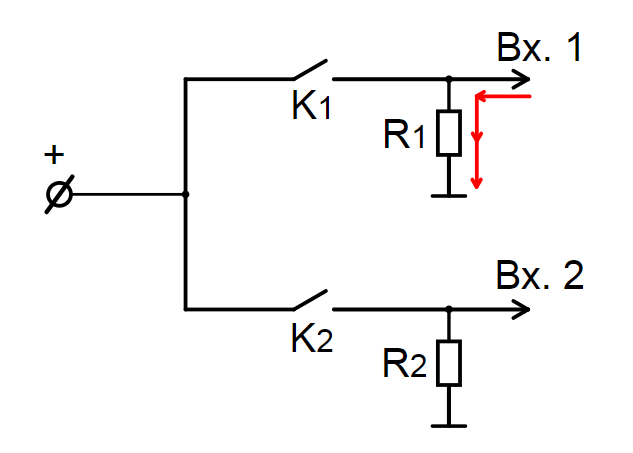


Рисунок . Функція підтягнення

Отже всі контакти, використані для введення сигналу, мають бути підтягненими.

### Сервер

Через те, що всі модулі ESP8266 з’єднані однією мережею Wi-Fi, кожному присвоюється свій ідентифікатор (IP-адреса).

Приймаючи натискання кнопок від пультів, сервер запам’ятовує і надісланий номер кнопки, і кожен такий ідентифікатор, щоб уникнути повторів: одна людина може відправити 100 натискань, кардинально змінюючи статистику, що призведе до хибного трактування результатів доповідачем. Отже, приймаючи чергове повідомлення від пульта, сервер повинен перевірити його ідентифікатор, і тільки якщо такого ще не має у списку, відповідь зараховується.

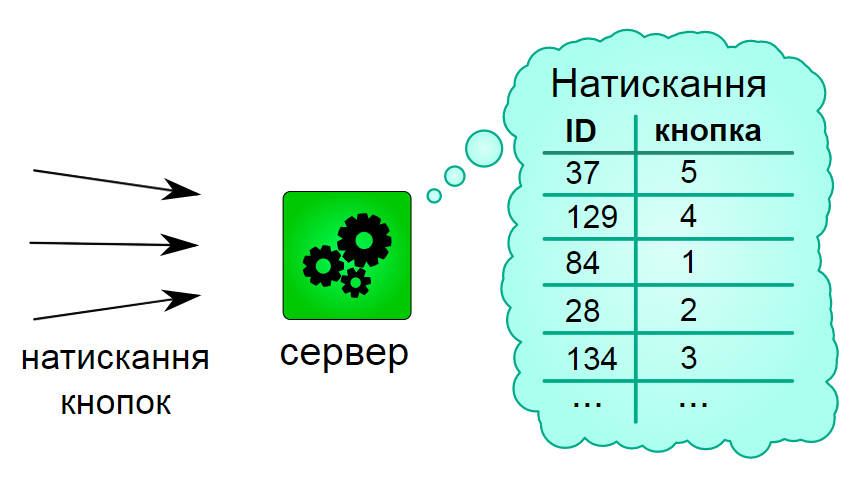


Рисунок . Пам'ять сервера

Порядок зв’язку сервера з ПК простий: ПК надсилає серверу команди («отримати результати» або «забути поточні результати»), а сервер надсилає ПК результати (натискання кнопок), якщо була відповідна команда.

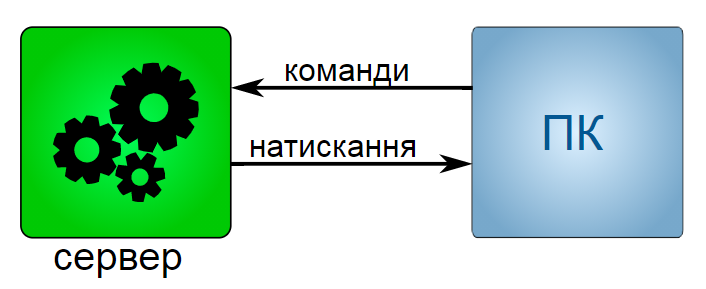


Рисунок . Обмін інформацією між сервером та ПК

## Створення програмного забезпечення

### Для ПК

Основною функцією ПЗ на ПК доповідача є прийняття результатів із сервера та відображення діаграми поверх вікна слайд-шоу Microsoft PowerPoint.

Організація вікон за Z-координатою у будь-якому віконному менеджері (в тому числі, і у Microsoft Windows) представлена чергою. За звичайних умов вікна переміщаються в початок (поверх інших), коли на них фокусується користувач. Але деякі вікна завжди залишаються на початку черги, як-от вікно слайд-шоу. Називатимемо таке вікно *закріпленим*.

Щоб інша (відмінна від PowerPoint) програма мала смогу вивести вікно над закріпленим вікном слайд-шоу, таке вікно також повинно бути закріпленим. Але порядок закріплених вікон також може змінюватися користувачем під час фокусування, або іншими чинниками, такими, як порядок запуску програм.

Отже використання самостійної, відмінної від PowerPoint програми для виведення діаграми є недоцільним. Тому діаграму має створювати сам PowerPoint.

Потрібного для даної задачі функціоналу в PowerPoint не існує. Для розширення його можливостей я створив власну *надбудову* (англ. Add-on) – плагін, що виконує сам PowerPoint, і який має змогу виводити будь-які вікна поверх слайд-шоу так, що користувач не зможе змінити їх розташування.

Надбудови всіх програм Microsoft Office являються проектами на скриптовій мові VBA (англ. Visual BASIC for Applications), що створюються та змінюються редактором, вбудованим у кожну таку програму. Все, пов’язане із надбудовами, розташоване на вкладці «Розробник», що за замовчуванням прихована.

До переваг надбудов можна також віднести можливість інтегрувати їх налаштування в стрічку інструментів PowerPoint, так що ніяких окремих файлів конфігурації користувачу редагувати не потрібно.

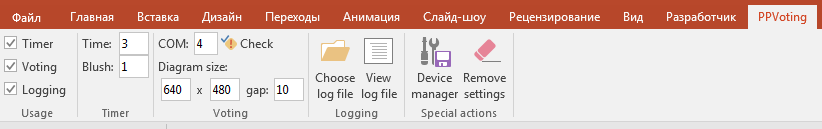


Рисунок . Інтегровані в PowerPoint налаштування надбудови

Свій проект надбудови я назвав PPVoting від англ. PowerPoint Voting – голосування у PowerPoint.

PPVoting складається з трьох компонентів:

* *Voting* – виконує поставлену вище задачу;
* *Timer* – виводить таймер в кутку вікна слайд-шоу, що веде відлік до кінця презентації; потреба в таймері може виникнути, коли презентація достатньо довга, і в ній багато запитань, тому доповідачу потрібно буде чіткіше слідкувати за часом;
* *Logging* (англ. ведення журналу) – виконує досить тривіальну задачу: виводить у файл (журнал) всю інформацію про виконання програми (помилки, попередження, а також отримані результати).

Будь-який компонент можна вмикати або вимикати в налаштуваннях, тоді надбудова відповідно застосовуватиме або ігноруватиме вказаний компонент.

#### Voting

Стандартна бібліотека VBA не впроваджує інструментів для роботи з послідовним портом USB. Для цього я використав сторонню бібліотеку CommIO [1]. Проте вона не підтримує стандарт VBA 7 (останній стандарт мови), та в ній введено обмеження на використання не більш як чотирьох портів USB (тобто, якщо під’єднати сервер до порта USB з номером 21, бібліотека викине помилку при спробі до нього під’єднатися). Все це я виправив самостійно.

Передача результатів на ПК повинна слідувати якомусь формату (бінарному або текстовому). Я обрав текстовий формат JSON (англ. JavaScript Object Notation) через його розширюваність, швидку змінюваність та просте використання у VBA. У стандартній бібліотеці для роботи з JSON також немає інструментів, тому я використав сторонню бібліотеку VBA-JSON [2].

Показування діаграми повинно керуватися доповідачем, при чому доповідач може користуватися презентером – пультом дистанційного керування, що імітує натискання «гарячих клавіш» слайд-шоу. Тому я вирішив пов’язати діаграму з натисненням лат. букви W або «,» (коми), що зазвичай роблять екран білим. На білому екрані й відображатиметься діаграма. Якщо ж доповідачу буде потрібно сховати екран, у нього є альтернатива: зробити його чорним лат. буквою B або «.» (крапкою).

#### Timer

Таймер показує поточний час та час до кінця доповіді; за заданий час до кінця тло таймера змінює колір з білого на рожевий.

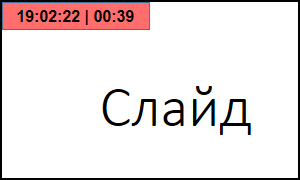
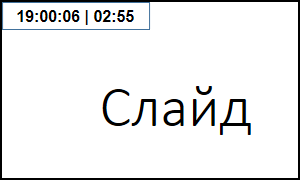


Рисунок . Таймер

Таймер повинен постійно оновлюватися, але це стає проблемою, коли у презентації застосовані плавні переходи між слайдами: слайд-шоу «мерехтить». Діло в тому, що для оновлення часу слайд-шоу повинно знаходитися на тому слайді, на якому малюється таймер в даний момент. Але при плавному переході логічно поточний слайд залишається тим самим, хоча практично починає відображатися наступний. Тому після того, як перехід закінчений на 50%, слайд-шоу на долі секунди вертається повністю до попереднього слайду, оновлює таймер та продовжує перехід до наступного.

Цей неприємний ефект я подолав таким чином: плагін постійно перевіряє стан кнопок, що відповідають за перемикання слайдів, і якщо хоча б одна натиснута, то плагін зчитує тривалість переходу для поточного слайда та зупиняє оновлення таймера на цей час.

### Для сервера та пультів

Як вже написано, у мережі Wi-Fi кожен пристрій має свій ідентифікатор – IP-адресу. Присвоєння IP може відбутися двома шляхами:

* + - 1. Автоматично – сервер сам підбирає ще не використаний IP та надсилає його пульту. Цей метод довший, особливо для великої кількості пультів.
      2. Статично – кожен пульт містить у своїй пам’яті свою адресу та надсилає її серверу, щоб той не підбирав нову. Цей метод швидший, тому в моєму проекті доцільніше використати саме його.

Головною проблемою програмного забезпечення є збереження налаштувань Інтернет-мережі та перехід в режим їх запису. Сервер та пульти голосування це роблять по-різному:

**Сервер** одразу після увімкнення надсилає запит на команду входу в режим запису налаштувань, після чого чекає кілька секунд та перевіряє, чи не надійшла йому відповідь. Якщо ж надійшла, він переходить в режим налаштувань.

**Пульт** одразу після увімкнення перевіряє, чи не натиснута середня кнопка. Переходить в режим налаштувань, якщо натиснута.

*Режим налаштувань* – інтерактивний режим, при якому пристрій отримує команди запису окремих налаштувань. Наприклад, послідовність команд r si192.168.1.8 w читає (r - read) налаштування з енергонезалежної пам’яті в оперативну пам’ять, встановлює (s - set) IP-адресу (i - ip) у значення 192.168.1.8 та записує (w - write) налаштування з оперативної пам’яті в енергонезалежну.

Код, реалізуючий режим налаштувань, один і той самий для обох пристроїв, тому входить до складу обох програм. Програми створюються за допомогою Arduino IDE (середовища, сумісного з ESP8266), що:

1. вимагає, щоб кожна програма знаходилася в окремій папці;
2. не допускає, щоб програма включала файли, які знаходяться за межами папки.

Тому спільний код налаштувань винесений в окрему папку, а після зміни слід копіювати файли в папки з кодами сервера та пультів.

## Створення засобів автоматизації

#### Налаштування сервера та пультів голосування

Єдиний процес, що потребує автоматизації – це запис налаштувань у сервер та пульти. Для цього я створив відповідні скрипти мовою Python, що автоматично з’єднуються з пристроєм, записують до нього програмне забезпечення, вводять його в режим налаштувань, записують всі налаштування та виходять із режиму, лишаючи пристрій повністю готовим.

Важливо зауважити, що скрипти потрібно відредагувати перед використанням. Найважливішим параметром є назва порту, до якого під’єднаний пристрій для налаштування.

Як і з VBA, у стандартній бібліотеці Python немає інструментів для роботи з послідовними портами та ESP8266. Для цього слід встановити додаткові бібліотеки. Скрипти автоматично перевіряють, чи не встановлені вже ці бібліотеки, та, якщо не встановлені, пропонують шляхи встановлення.

#### Хостинг проекту

Для простого перегляду, ознайомлення та використання мого проекту, я опублікував його на популярному сайті хостингу програмних проектів GitHub [3]. Будь-який користувач Інтернету може отримати копію програмного коду, щоб самостійно скомпілювати, або завантажити готові бінарні файли.

Особливістю GitHub є використання файлу у форматі розмітки Markdown (зазвичай, README.MD), що знайомить з проектом та виступає в ролі документації.

# Висновок

Я створив систему бездротового пультового голосування, інтегровану в програму Microsoft PowerPoint. Найкраще така система підходить для аналізу якості сприйняття інформації аудиторією та для скринінгового бліц-опитування, коли користуватися власними гаджетами учасникам заборонено.

Під час виконання свого проекту я оволодів такими технологіями та знаннями:

* мовою програмування VBA;
* інструментами розробників для Microsoft Office;
* протоколами Wi-Fi, HTTP та форматом збереження даних JSON;
* особливостями роботи мікроконтролера ESP8266;
* особливостями програмування мікроконтролерів мовою C++ у середовищі Arduino IDE;
* системою контролю версій Git;
* онлайн-сервісом хостингу проектів GitHub;
* мовою розмітки Markdown для опису та документації проектів,

також я здобув такі навички:

* конструювання електронних схем за допомогою онлайн-сервісів на основі знань фізики;
* написання та налагодження програмного коду;
* створення комп’ютерних систем;
* автоматизації своєї роботи за допомогою скриптових мов програмування;
* виявлення недоліків сторонніх програм та їх обхід або виправлення,

і до того ж я поглибив свої знання мов програмування C++ та Python.

# Джерела

|  |  |
| --- | --- |
| [1] | D. M. Hitchner, «CommIO,» [Онлайновий]. Available: http://www.thescarms.com/VBasic/commio.aspx. [Дата звернення: 02 Травень 2021]. |
| [2] | T. Hall, «VBA-JSON,» [Онлайновий]. Available: https://github.com/VBA-tools/VBA-JSON. [Дата звернення: 02 Травень 2021]. |
| [3] | Л. М. Дмитрович, «PPVoting,» [Онлайновий]. Available: https://github.com/MarkLagodych/PPVoting. |