

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.

« ____ » _____ 20__р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА
ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Інформаційне табло на базі світлодіодного дисплею

Освітньо-кваліфікаційний рівень "бакалавр"
Напрямок підготовки 6.050102 "Комп'ютерна інженерія"
(шифр і назва спеціальності)

Керівник проекту:

(підпис)

Кардашук В.С.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Хищев В.О.

(ініціали, прізвище)

Група:

КІ-14аД

Севєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 6.050102 "Комп'ютерна інженерія"

(шифр і назва)

Спеціальність _____

(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____

_____ І.С. Скарга-Бандурова

« _____ » _____ 20 ____ р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) БАКАЛАВРА**

Хишева Владислава Олександровича

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційне табло на базі світлодіодного дисплею

керівник проекту (роботи) к.т.н., доц. Кардашук В.С.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 14.05.2018 р. № 117/48

2. Термін подання студентом роботи 11.06.2018

3. Вихідні дані до роботи матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
електронні плакати

РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра: __ с., __ рис., __ табл., __ бібліографічних джерел посилань, __ додатків.

Об'єкт розробки: програмно-технічні засоби для виготовлення інформаційного табло на базі світлодіодного дисплею.

Мета роботи: розробка макету інформаційного табло для міської зупинки громадського транспорту.

В проекті виконано:

- 1) виконано огляд існуючих технологій моніторингу;
- 2) сформульовані необхідні вимоги щодо розроблюваного пристрою;
- 3) розроблено алгоритм прогнозування часу прибуття транспортного засобу;
- 4) розроблено веб-додаток для отримання прогнозу;
- 5) розроблено макет пристрою;
- 6) виконано збірку пристрою;
- 7) виконати розрахунки по розділу «Охорона праці».

Отримано наступні результати: макет інформаційного табло виводить час прибуття тролейбуса на зупинки міста, завдяки прогнозу, а не згідно розкладу.

Практичне значення, галузь застосування роботи: транспортна система міста.

Ключові слова: інформаційне табло, час прибуття, світлодіодний дисплей, прогноз, програмно-технічні засоби.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєвєродонецьк, 93400.

ЗМІСТ

ВСТУП.....	5
1.ОГЛЯД ІСНУЮЧИХ ПРОБЛЕМ В ОБЛАСТІ МОНІТОРИНГУ ТРАНСПОРТУ	6
1.1 Актуальність проблеми громадського транспорту.....	6
1.2 Огляд технологій моніторингу громадського транспорту в режимі реального часу.....	7
1.3 Огляд існуючих систем моніторингу громадського транспорту	17
1.4 Постановка завдання	25
2. РЕАЛІЗАЦІЯ ПРОГРАМНИМИ ЗАСОБАМИ РОБОТИ ПРИСТРОЮ.....	26
2.1 Бібліотека сURL	28
2.2 Протокол HTTP	28
2.3 Формат обміну даними JSON	31
2.4 Синхронізація часу за рахунок NTP серверу	34
2.5 Протокол для взаємодії Інтернет Речей.....	38
3 РОЗРОБКА ІНФОРМАЦІЙНОГО ТАБЛЮ НА БАЗІ СВІТЛОДІОДНОГО ДИСПЛЕЮ.....	43
3.1 Модулі для зборки	43
3.2 Модулі для зборки макету	46
3.3 Блок живлення.....	47
3.4 Контролер	47
3.4 Архітектура системи.....	49
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	54
4.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу	54
4.2 Електробезпека	55
4.3 Освітлення.....	57
4.4 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	60
Висновок	65
ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	68
ДОДАТОК А. Слайди презентації.....	71
ДОДАТОК В. Код скетчу.....	78

ВСТУП

На сьогоднішній день в світі з неймовірною швидкістю розвиваються технології, які в свою чергу тягнуть за собою стрімку інтеграцію в усі інші галузі життєдіяльності людини, тому, будь-яка сфера господарства є синтезом різних наук. Таким чином, пріоритетним завданням є соціальний вплив на поліпшення комфортабельності проекту «InTab», а також удосконалення технологій, конфігурацій громадського транспорту. Подібні системи дають можливість бути проінформованим про наближення всіх можливих міських видів транспорту, про приблизний час прибуття автобуса чи тролейбуса, що значно покращує якість споживання міським громадським транспортом. А також «InTab» послужить хорошим помічником для туристів з-за кордону в вивченні та орієнтуванні в місті.

У цьому дипломному проекті розглядаються питання розробки інформаційного табло для програмного забезпечення, що здійснює моніторинг громадського транспорту в режимі реального часу.

Дипломний проект складається з вступу, чотирьох розділів, висновків та двох додатків.

1 ОГЛЯД ІСНУЮЧИХ ПРОБЛЕМ В ОБЛАСТІ МОНІТОРИНГУ ТРАНСПОРТУ

1.1 Актуальність проблеми громадського транспорту

Сучасна тенденція розвитку регіону призводить до збільшення кількості транспортних засобів у мережах дорожнього руху. Внаслідок чого зростає потреба в організації раціонального підходу транспортних потоків. До того ж саме рівень розвитку транспортної системи є однією з головних ознак економічної стабільності та процвітання міста. Єдина транспортна система має забезпечувати розвиток і функціонування всіх видів транспорту, та максимально відповідати потребам міста при мінімальних витратах. Все це, суттєво, сприяє економічному, соціальному і культурному розвитку суспільства. Апаратно-програмний комплекс «InTab» - це перший крок для підвищення рівня доступності та поліпшення якості обслуговування в громадському транспорті міста. Проект «InTab» спрямований, перш за все, на збереження вільного часу населення. Не секрет, що у переважної більшості людей немає бажання витратити марно свій вільний час, у тому числі при очікуванні громадського транспорту на зупинках міста. Набагато зручніше мати інформацію про те, у який час і за яким маршрутом здійснюється рух транспортного засобу. Даний проект сприяє вирішенню цієї проблеми, шляхом розробки апаратно-програмного комплексу, що побудований по технології IoT і призначений для інформування про час прибуття громадського транспорту на зупинки міста. Доречними є також безперервний автоматичний збір, збереження навігаційної інформації про місцезнаходження транспортного засобу, завдяки чому можливо здійснення управління рухом і внесення своєчасних коригувань диспетчерськими службами.

1.2 Огляд технологій моніторингу громадського транспорту в режимі реального часу

1.2.1 Глобальні навігаційні супутникові системи

В даний час широко використовуються дві глобальні супутникові навігаційні системи на базі диференціальних підсистем ГЛОНАСС (Глобальна навігаційна супутникова система) і NAVSTAR (Navigation system with timing and running - Навігаційна система з визначенням часу та дальності - США). У сучасному світі NAVSTAR більш відома як GPS (Global Position System - Система Глобального Позиціонування). Дані системи дозволяють позиціонувати необхідні координати з точністю до декількох сантиметрів. Також ведуться розробки аналогічних систем GALILEO (Європейський Союз) і Compass (Китай). Технологія супутникової навігації використовується в багатьох сферах нашого життя, як в науці, так і в побуті. Завдяки їй відбувається виявлення географічних координат (ширина, довжина, висота над рівнем моря) різних об'єктів при будь-яких умовах.

Високі експлуатаційні характеристики на структурному рівні здійснюються шляхом спільного функціонування основних сегментів, які будуть розглянуті далі.

Функціональні сегменти

Виділяють три основних функціональних сегмента:

- 1) космічний сегмент;
- 2) сегмент управління;
- 3) сегмент споживачів.

Системи ГЛОНАСС і GPS повинні забезпечувати шукані дані безпосередньо в апаратурі споживача. Внаслідок чого відбувається ускладнення обладнання споживача. Всі обчислення відбуваються тільки на основі пасивно прийнятих сигналів від навігаційного космічного апарату (НКА) з задалегідь

точно відомими орбітальними координатами.

1) Космічний сегмент

Забезпечує точність визначення місця розташування і стабільність функціонування, яка залежить від орбітального розташування супутників і параметрів їх сигналів. Необхідно, щоб в зоні видимості об'єкта знаходилося від 3 до 5 супутників, які знаходяться на висоті близько 20 000 км. На даній висоті можна прийняти сигнали кожного НКА майже на половині поверхні Землі, що дозволяє забезпечити безперервність радіонавігаційного поля.

Сигнали, які випромінюють НКА, мають дві складові - далекомірну і службову. Далекомірна необхідна для визначення просторової орієнтації, а службова зберігає інформацію про шкалах часу, координатах супутників і векторах швидкостей НКА.

Варто відзначити, що далекомірна складова впливає на точність, яка поділяється на стандартну, доступну всім, і високу, до якої мають доступ лише військові організації. Цей поділ здійснюється за рахунок кодування сигналу високої точності.

2) Сегмент управління

Складається з головної станції, контрольно-вимірювальних станцій (КІС), пов'язаних з головною станцією і між собою каналами зв'язку; наземного еталона часу частот.

КІС здійснює первинну обробку інформації і виробляє обмін даними з головною станцією, де відбувається збір даних з усіх КІС і їх аналіз з наступним завантаженням в бортову ЕОМ. Дані поділяються на оперативні та довгострокові.

Наземний еталон часу і частоти має більш високу точність, ніж бортові. За допомогою нього відбувається синхронізація і корекція бортових процесів, еталонів.

3) Сегмент споживачів

Поділяються на військові / цивільні організації і на приватних осіб.

Одиниці мір часу

Для зручності були введені дві системи відліку часу: астрономічні; неастрономічні.

До астрономічних відносяться зоряна доба (зоряний період обертання Землі, сонячна доба).

Зоряною добою називається проміжок часу, відміряний між двома послідовними верхніми кульмінаціями точки весняного рівнодення. Сонячними цілодобовими називають інтервал часу, відрахований з нижнім кульмінаціям центра видимого диска Сонця (істинного Сонця).

В силу того, що добове обертання Землі нерівномірно, відбувається невелика зміна зоряної доби. Внаслідок цього факту для виконання точних розрахунків було введено рівномірно поточний ефемеридних час ET (по ньютонівській механіці). За одиницю часу було прийнято ефемеридна секунда, яка визначається, як $1/86400$ частка середньої тривалості доби в 1900 році.

Неастрономічний час було введено для зручності визначення місцеположення супутників, прив'язці їх сигналів в єдиній шкалі. Для зручності споживачів і сегмента управління вводять різні системи відліку.

1) Всесвітній час UT (Universal Time)

Містить рік, місяць, число, час, хвилину, секунду.

2) Всесвітній координований час UTC

Цей час вимірюється аз атомним годинником. Показання даного годинника мають період від 0.5 до 2.5 секунд. І мають таку корекцію на 1 секунду, щоб $UT-UTC < 0.9$ с. Так як в супутникової навігації використовується атомний стандарт частоти, то при синхронізації UTC відбувається корекція, внаслідок чого виникає секундний стрибок.

Для порівняння: 19:21:01 по UT на GMT +00: 00 = 22:21:01 на GMT +03: 00.

В якості еталонного годинника використовують випромінювання цезієвого атомного променевого стандарту. Секунда визначена як 9192631770

періодам випромінювання атома цезію-133.

Необхідно відзначити відмінності між GPS і ГЛОНАСС. Система GPS для визначення положення супутників в певний момент часу в просторі користується показанням годин системи GPS на даний момент. У той час як ГЛОНАСС використовує свідчення годин московського дискретного часу, зміщеного відносно показань годин системи ГЛОНАСС.

Шкали часу і їх синхронізація

Так як навігаційна система складається з безлічі контрольних станцій, супутників і приймачів, які повинні бути з жорстко фіксованими просторово – часовими параметрами, вводять тимчасові шкали для синхронізації всіх компонентів супутникової навігації.

Виділяють три шкали часу - системну, бортову і шкалу споживача.

Основна є системна шкала. За допомогою неї відбувається синхронізація всіх підсистем СНС. Для її реалізації використовують найбільш точні квантові еталони часу і частоти, розташованих на КІС і забезпечені спеціальними технічними і алгоритмічними рішеннями.

Бортова шкала заснована на «атомних годинниках» (квантовим еталоном), які знаходяться на борту НКА. Відповідно, всі сигнали, що випромінюються супутником, спираються на бортовий еталон.

Варто відзначити, що бортові еталони мають деяку розбіжність з системними, через те, що працюють в більш жорстких умовах і мають менше технічне забезпечення. Тому наземний комплекс управління звіряє ці шкали за формулою:

$$t_{БШВ} = t_{изл} + \Delta t_{распр} + \Delta t_{рз} + \Delta t_{ам} + \Delta t_{анн}, \quad (1.1)$$

де $t_{изл}$ - момент початку випромінювання сигналу за бортовою шкалою, $\Delta t_{распр}$ - час поширення фазового центру антени, що передається НКА до фазового центру прийомної антени вимірювальної станції $\Delta t_{рз}$ - вихід бортової

шкали часу з релятивістських ефектів, які визначаються з точністю до нс, Δt_{am} - зрушення бортовий шкали на величину затримки поширення в атмосфері, Δt_{ann} - апаратурні і метрологічні похибки. Вихід бортової шкали часу щодо системної визначається різницею між обчисленим значенням і еталонним часом системи.

Виділяють два методи вимірювання дальності до НКА для обчислення $\Delta t_{распр}$ активний (запитальний) і пасивний (беззапитний). Активний метод вважається більш точним, але для його реалізації необхідна додаткова апаратура.

Система ГЛОНАСС використовує активний, а GPS - пасивний метод.

У разі відхилення бортовий шкали від системної на значення, більше допустимого, відбувається корекція. Вона буває двох видів.

При зміщенні тимчасових інтервалів шкал (різниця фаз кодових послідовностей) проводиться фазування бортовий шкали, яке дозволяє поєднати шкали з точністю до десятків наносекунд. Спільно з фазування або незалежно від нього може проводитися зміна оцифровки (корекція коду на ціле число секунд).

В даний час корекцію бортових шкал виробляють епізодично, в силу того, що всі супутники управляються з обмежених територій. Тому застосовується прогнозування догляду бортовий шкали. Ідея полягає в тому, що після тривалих спостережень за бортовою шкалою створюється математична модель і визначається систематична складова догляду. В наслідок завдяки даній складовій в бортову пам'ять кожного супутника завантажуються частотно-часові поправки, які пізніше передаються споживачеві в складі ефемеридної інформації.

Шкала споживача є найбільш нестабільною і зміщеною щодо системної шкали. Вона коригується за допомогою деякої інформації, прийнятої з супутника. Існує три способи для синхронізації шкал.

В першому випадку інформація, отримана споживачем, використовується для розрахунку поточного відхилення бортовий шкали часу щодо системної і

для прив'язки шкали часу споживача до системної шкалою при знаходженні тимчасової координати V . Цей спосіб забезпечує точність прив'язки не менше 1 мкс і частіше застосовується.

Інший спосіб заснований на тому, що споживачеві в навігаційному повідомленні передаються, мітка часу НКА і частотно-часові поправки. Точність прив'язки в цьому випадку нижче і залежить від знання дальності від НКА ($\Delta t_{распр}$).

Третій спосіб аналогічний другому, але в якості джерела інформації використовується дальномірні коди.

1.2.2 Роль технологій супутникової навігації ГЛОНАСС / GPS в системах моніторингу

Позиціонування рухомого об'єкту, переміщення і точного часу з використанням технологій супутникової навігації ГЛОНАСС / GPS знайшло широке застосування в системах моніторингу транспорту. Зараз застосування технологій автоматизованого супутникового спостереження і контролю - незамінна складова, з метою якої підняти процес управління автопарком на новий ефективний рівень. Спочатку системи моніторингу застосовувалися виключно для контролю / моніторингу переміщення транспортних засобів (ТЗ) і функціонували тільки в офлайн режимі, не дозволяючи в режимі реального часу стежити за об'єктом.

З розвитком технологій передачі даних GSM / GPRS, а також web-технологій, системи моніторингу автотранспорту дозволили здійснювати дистанційне спостереження за транспортом цілодобово, в режимі реального часу. У системах моніторингу та контролю транспорту використовується поєднання навігаційних і телекомунікаційних технологій. Наприклад, в якості

каналів передачі даних в основному використовується GSM / GPRS, а у віддалених районах, де відсутнє покриття стільникового зв'язку - системи супутникового зв'язку (ССС). На підприємствах і в відомствах, в яких необхідні додаткові резервні канали, в якості резервних (а іноді і основних) каналів використовується Конвенціональний і транкінговий радіозв'язок. Подібна система представлена на рисунку 1.1.

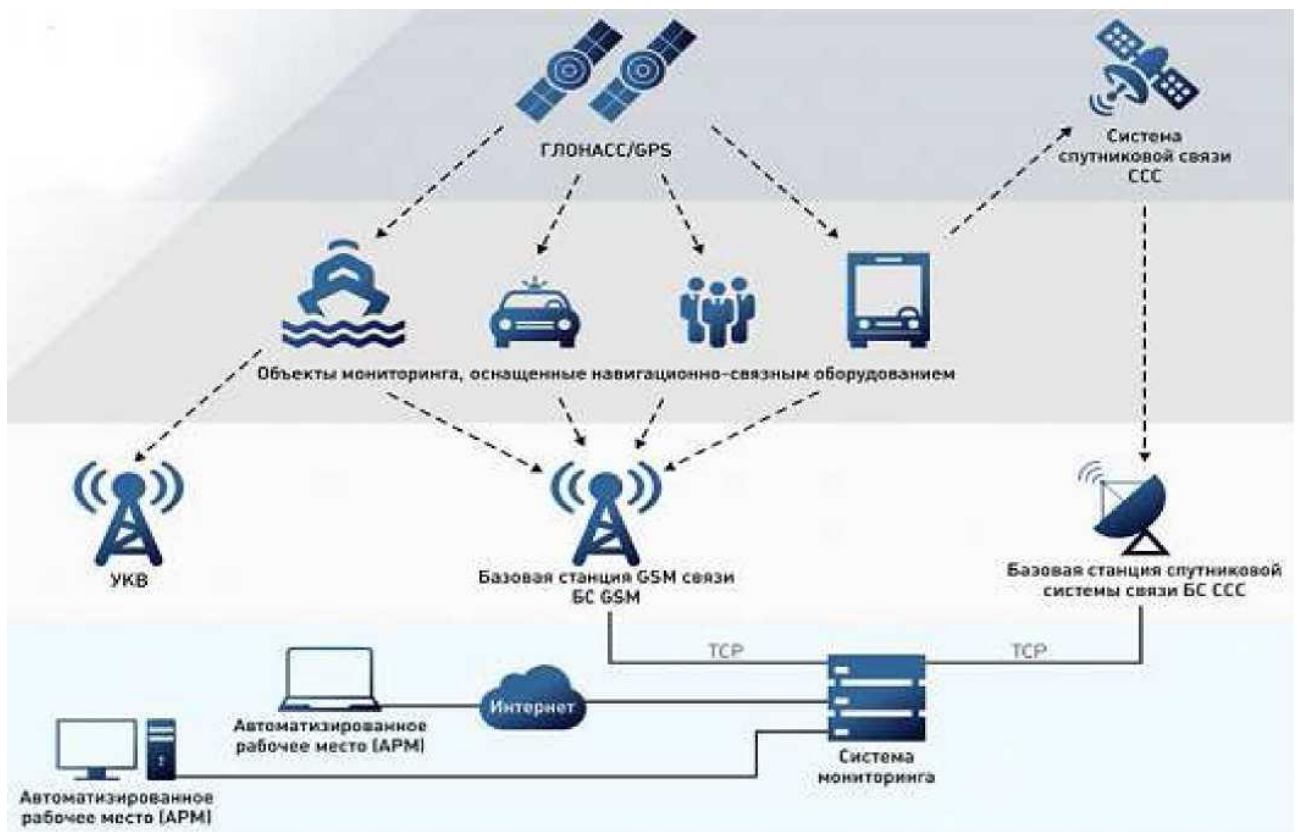


Рисунок 1.1 - Схема моніторингу транспорту в режимі реального часу

Загальний принцип роботи системи такий, що на транспорт встановлюється спеціальне бортове навігаційне обладнання моніторингу транспорту, в який вбудований ГЛОНАСС / GPS-приймач, що обробляє сигнали зі супутникових навігаційних систем.

Ці пристрої зазвичай називають абонентськими терміналами. І зараз виробники роблять їх багатосистемними, щоб поліпшити якість прийому

сигналу. Так само в термінали встановлюється SIM карта стільникового оператора, за допомогою якої відбувається передача даних в диспетчерський центр.

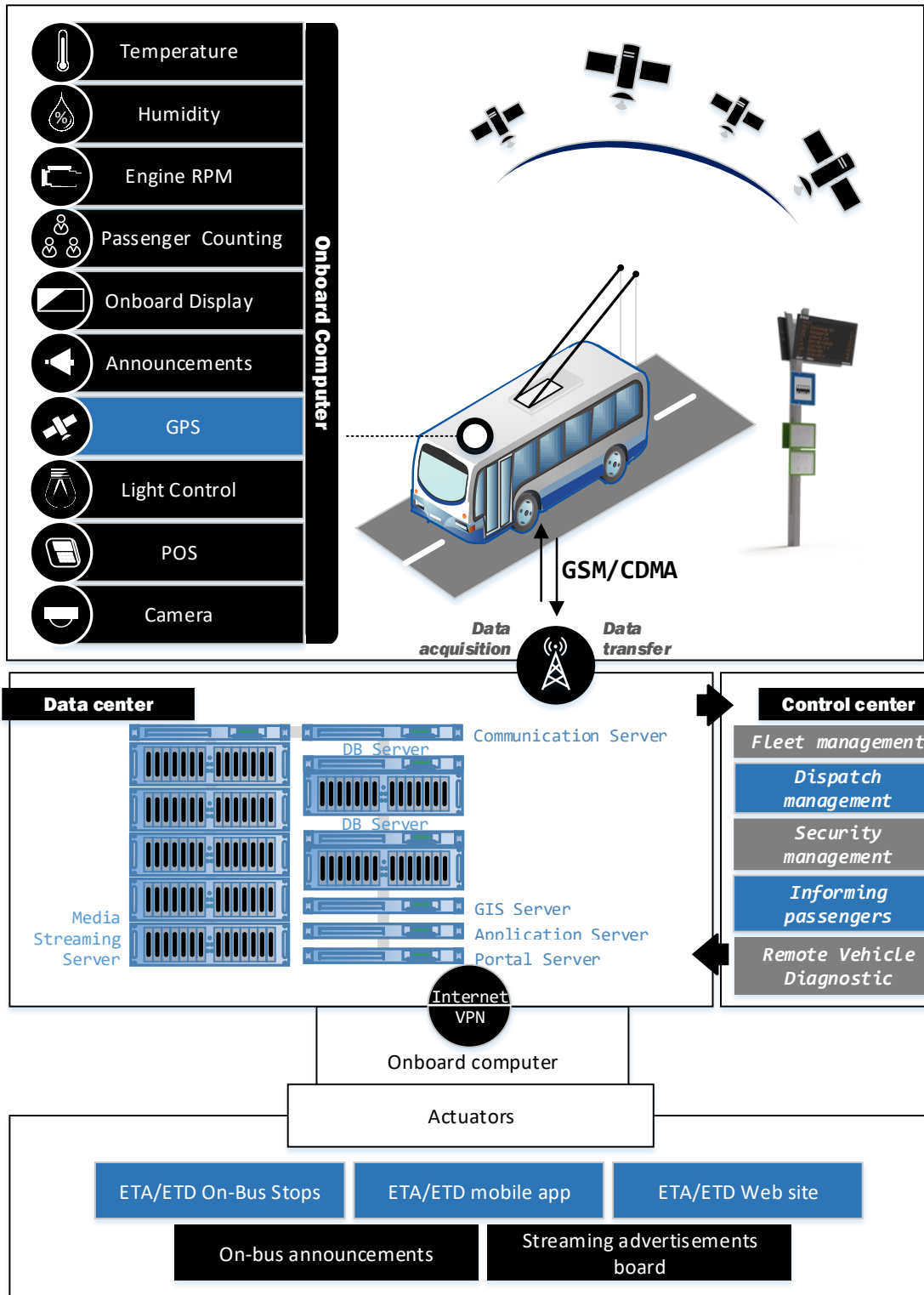


Рисунок 1.2 – Інфраструктура ІТС

Розвиток технологій дозволило так само використовувати в системі моніторингу телеметрію об'єктів. До терміналів, в даний час можливе підключення різних датчиків і виконавчих пристроїв, які дозволяють або подавати водієві сигнали тривоги в режимі реального часу («тривожна кнопка»), або відстежувати заправки і зливи палива (датчик рівня палива), або датчики стану механізмів, до наприклад - вимірювачі температури, кінцеві вимикачі (датчики відкриття дверей і капота, підйому кузова) і т.д.

Причому телеметричну інформацію можна отримувати як зі штатних датчиків (через CAN інтерфейс), так і з додатково встановлених. Останнім часом стали популярні і все частіше використовуються засоби фото - і відеофіксації (камери всередині кабіни і салону і т.д.). Отже, термінал отримує навігаційні сигнали, збирає телеметричні дані з додаткових і штатних пристроїв і далі передає їх на телематичний сервер через комунікаційне середовище.

На сервері вся інформація обробляється і далі відправляється диспетчеру або оперативному черговому на автоматизоване робоче місце (АРМ). АРМ - це комп'ютер з доступом в Інтернет і з організацією роботи через спеціальну клієнтську програму - програмне забезпечення для автоматизації роботи диспетчера або оперативного чергового. Сьогодні системи моніторингу транспорту - це вже не просто спостереження за транспортом на електронній карті, це вже багаторівневі територіально - розподілені системи, що володіють не тільки функціями моніторингу, а й управління, в тому числі глибокого аналізу та обліку.

На сьогоднішній день системи моніторингу автотранспорту на основі ГЛОНАСС і GPS - це ефективний інструмент управління автопарками, забезпечення безпеки користувачів транспортних засобів, мінімізації витрат на експлуатацію транспорту, зростання прибутку підприємства. Ефективність впровадження системи моніторингу транспорту виражається в наступних показниках:

- 1) мінімізація збитків від крадіжок вантажу, викрадень транспортного

засобу і його нецільового використання;

2) мінімізація витрат на технічне обслуговування і пально-мастильні матеріали за рахунок оптимізації маршрутів і зниження непродуктивного пробігу автотранспорту;

3) зниження потреби в розширенні парку автотранспорту;

4) підвищення транспортного обслуговування клієнтів і можливість залучення нових клієнтів за рахунок розширення спектра послуг і оперативного реагування на запити;

5) оптимізація планування роботи на основі об'єктивної інформації про реальний пробіг автотранспорту і зниження витрат, пов'язаних з його ремонтом і простоєм;

б) підвищення ефективності роботи персоналу і можливість введення системи матеріального стимулювання, що базується на достовірній інформації про роботу кожного водія і заохочує більш ефективне використання робочого часу, транспорту, паливно-мастильних матеріалів та спеціального обладнання і т.д.

Система передачі та обробки сигналу має послідовність. Сигнал, який передається з супутника на транспортний засіб за допомогою радіосигналів і Інтернету (протокол TCP / IP), надходить на телематичний сервер. З сервера же сигнали надходять в диспетчерський пункт, де інформація обробляється і виводиться на інформаційне табло, встановлене на зупинці, залежно від запитуваної інформації вона розходить по різних міських зупинкам.

Інформаційні табло для зупинок громадського транспорту, інформативно і зручно в користуванні.

Табло відображає: номер маршруту, час прибуття маршруту, поточний час, температуру і дату.

Табло може розміщуватися на окремій опорі в зоні зупинок, кріпиться на стовпі міського освітлення, або встановлюється всередині павільйону на зупинці.



Рисунок 1.3 – Реальний аналог інформаційного табло

1.3 Огляд існуючих систем моніторингу громадського транспорту

На сьогоднішній день всі сфери людської життєдіяльності розвиваються з величезною швидкістю, одним з основних видів розвитку і поліпшення робочих структур є їх автоматизація. Зараз автоматизації піддається практично кожна системна структура. Процес автоматизації не обійшла стороною і міські автобусні зупинки, так як все це робиться заради зручності та оптимізації роботи, а також для соціального поліпшення міста. Аналоги подібної системи існують в багатьох країнах, перші подібні смарт-зупинки були введені в Чехії, точніше в Празі. Суть роботи подібної системи полягає в поліпшенні соціального вигляду міста, відстеження міського транспорту в режимі реального часу. Розробкою даних смарт-зупинок займаються вже близько 10 років.

Лондон. TfL запускає новий екран з дисплеєм на шині на батарейках, який вперше надасть інформацію про подорожі в режимі реального часу в автобусних службах.

Перше випробування проводиться на автобусній зупинці в Нортвуд-стріт, Хіллінгдон, і надасть клієнтам можливість подорожувати в лікарню Mount Vernon і з неї, щоб дізнатися всі останні дані про поїздку, які їм потрібні.

Нові екрани можна швидко і легко підключити до стовпів автобусної

зупинки і відобразити клієнтам інформацію про прибуття наступного автобуса. Панелі живляться від батареї, тому вони не обмежені зупинками автобусів з укриттями і електроживленням.

Перший з цих нових екранів буде встановлений на автобусній зупинці на зеленій смузі, на станції Northwood, Хіллінгдон. На екрані відобразиться інформація про автобусних службах TfL, що працюють від зупинки автобуса (маршрути 282 і N11), а також транскордонних служб, в даному випадку маршруту Arriva 8 (який проходить від Abbots Langley до Mount Vernon Hospital через Watford Junction).

У столиці близько 2,600 автобусних зупинок з дошками Countdown, які надають клієнтам інформацію про рух автобуса. Ця інформація також доступна на веб-сайті TfL і в різних додатках.

Також є 10 зупинок на TfL-мережі для пробних екранів E-paper, які відображають інформацію про шину, яка може бути змінена віддалено, а не звичайні розклади, які необхідно змінити вручну.

Інформаційна панель має розмір 458 мм (ш) x 375 мм (ч) x 68 мм (г). Партнером поставки є Axentia Technologies of Sweden <http://www.axentia.se/>.

Батареї, які живлять динамічні екрани, мають термін експлуатації три роки.

Сінгапур. Інформаційні панелі прибуття автобусів були запущені в 2007 році, до кінця року кількість панелей, які повідомляють пасажирів про прибуття, зросте більш ніж на 50 відсотків, а нові будуть встановлені на автобусних зупинках з декількома службами.

В даний час прибуття автобуса з'являється на панелях дисплея на 51 автобусній зупинці. До кінця цього року до схеми буде додано ще 31 зупинка в таких областях, як Ang Mo Kio, Commonwealth, Eunos, Tampines і Tiong Bahru.

Панелі оновлюються кожну хвилину, але ЛТА сказав, що фактичний час прибуття автобуса може варіюватися протягом трьох хвилин.



Рисунок 1.4 - Приклад інформаційної панелі Сінгапуру

Таллінн. Щоб зробити громадський транспорт більш привабливим, Таллінн переключиться на інформаційну систему в режимі реального часу.

На будь-якому каналі не було інформації в реальному часі про розклад. Крім того, інформація практично не була доступна в разі надзвичайних змін і затримок в графіках часу прибуття (екстремальні погодні умови, позачергові ремонтні роботи, непрацюючі автомобілі і т. п.). Надання додаткової інформації про фактичний статус громадського транспорту в режимі реального часу є одним із способів зробити розклад більш привабливим для користувачів.

Цей захід включало:

- 1) установку електронних дисплеїв в режимі реального часу на 6 зупинках;
- 2) створення додаткових функцій на веб-сайті, що містять інформацію про розклад;

3) створення мобільних додатків для смартфонів з використанням iOS і операційних систем Android для надання інформації в режимі реального часу.

Основними цілями заходу є:

- 1) створити інформаційну систему, орієнтовану на пасажирів, яка забезпечує надійну і актуальну інформацію про громадський транспорт;
- 2) покращити доступ та якість інформації про пасажирів на зупинках громадського транспорту;
- 3) підвищити задоволеність користувачів послугами громадського транспорту;
- 4) збільшити використання громадського транспорту.

Цей захід було реалізовано на наступних етапах:

Етап 1: Попередні дослідження (січень 2009 - квітень 2011 року) - дослідження інформаційних систем реального часу, оптимальних технічних рішень для інформаційної системи реального часу в Таллінні, визначення зупинок високої щільності;

Етап 2: Переговори з постачальниками (березень 2011 року - липень 2012 року). Переговори з постачальниками основних технічних рішень, дисплеїв, додаткових функціональних можливостей веб-сторінок і мобільних додатків;

Етап 3: Технічне проектування і впровадження рішень (грудень 2011 року - жовтень 2012 року) - створення додаткових функціональних можливостей веб-сторінки, розробка програмного забезпечення і мобільних додатків, проектування та встановлення електронних дисплеїв;

Етап 4: Вимірювання працездатності (грудень 2012 року). Дисплеї на зупинках стали повністю працездатними. Карта веб-сторінок з пунктами в реальному часі почала функціонувати в січні 2013 року. Станом на січень 2013 року запровадження мобільного додатка планувалося до кінця березня 2013 року.

Електронні дисплеї засновані на світлодіодним технології, в 6 найбільш підходящих місцях були обрані відповідно до пасажирських обсягів на

конкретних зупинках, де зупинки обслуговувалися лініями, які належать системі пріоритету (пріоритет транспортного засобу в світлофорах, що забезпечує кращу надійність до ліній), і де дисплеї потенційно можуть мати найбільший ефект для задоволення пасажирів. Розмір дисплея залежить від кількості ліній, які обслуговують зупинки. Дисплеї отримують інформацію про реальний час з центрального сервера за допомогою радіозв'язку, частота оновлення інформації в реальному часі становить 10 секунд.

Інформація, яка відображається на дисплеї зупинки:

1. Час;
2. Номер і номер лінії;
3. Час розкладу часу виходу наступного транспортного засобу з зупинки в форматі гг: мм, коли відправлення відправляється більш ніж на 1 годину, або конкретний транспортний засіб не входить в систему;
4. Розрахунковий час виходу наступного транспортного засобу з зупинки в хвилину, коли час прибуття менше 1 години, і коли конкретний транспортний засіб входить в систему;
5. Інформація про проблеми.

Люди з ослабленим зором можуть також отримувати звукову інформацію з дисплея за допомогою сумісних бездротових пристроїв. Веб-сторінка для розкладів знаходиться за адресою <http://soiduplaan.tallinn.ee/>. На сторінці представлена інформація про лінії, розклад, зупинки, місця розташування транспортних засобів, прогнозований час прибуття на зупинки, планування поїздки та інформації про квитки. В даний час частота оновлення даних (січень 2013) становить 30 секунд, але її планується скоротити.

Інформація в реальному часі, яка використовується в усіх трьох додатках, заснована на інформації GPS-позиціонування від транспортних засобів. Інформація відправляється з транспортних засобів на центральний сервер за допомогою радіозв'язку і може використовуватися для різних додатків і служб.

Вплив цього заходу було оцінено з урахуванням зміни інформованості і

вживання широкої громадськості планів впровадження електронних інформаційних систем в режимі реального часу на зупинках. Дослідження проводилися в 2009 і 2012 роках.

Ключовими результатами оцінки були:

1) підвищення обізнаності на 37%; - поінформованість про план впровадження електронних дисплеїв в режимі реального часу на автобусних зупинках збільшилася на 37%;

2) ухвалення збільшено на 8% - ухвалення необхідності відображення інформації в режимі реального часу в реальному часі на автобусних зупинках і можливий вплив їх з комфортом використання збільшився на 8%.

Крайова. Щоб більш ефективно координувати інформацію про трафік, Craiova розробляє інноваційну систему управління автопарком, що використовує відстеження транспортних засобів.

Для поліпшення інформаційних потоків будуть використовуватися розширені інформаційно-комунікаційні технології (ІКТ). Вісімдесят автобусів і 27 трамваїв будуть оснащені системою GPS, а 20 станцій будуть оснащені цифровими інформаційними панелями реального часу. Система управління трафіком буде пов'язана з центром управління трафіком, який буде представлений як частина заходів електронного квитка. Основна функція у всіх системах управління автопарком - відстеження транспортних засобів на основі GPS. Як тільки транспортний засіб розташоване, GPS передає інформацію про швидкість і напрям до прикладного програмного забезпечення. Це полегшує профілювання водія / транспортного засобу, профілювання проїзду (маршрут, швидкість, час, положення) і ефективність автомобіля.

Розширені системи управління автопарком підключені до бортового комп'ютера автомобіля і збирають дані для користувача. Деталі, такі як пробіг і витрата палива, збираються в глобальній схемою статистики. Система управління автопарком, яка буде застосовуватися в Craiova, буде складатися з обладнання для відстеження транспортних засобів, пакетів даних, програмного

забезпечення для настільних комп'ютерів, хостингу бази даних, технічної підтримки і панелей на найважливіших станціях з інформацією в реальному часі для пасажирів. Інструменти інформаційної мобільності будуть інтегровані в систему з двома основними функціями: перший - це моніторинг парку з автоматичним моніторингом автомобілів (AVM) на основі GPS, а другий - інформація про пасажирів з даними, наданими системою AVM. Цей захід буде реалізовано в чотири етапи: дослідження, демонстрації, навчання та оцінка.

Дослідження, пов'язане з обладнанням GPS / GPRS, було завершено, і технічні специфікації були додані в загальну тендерну книгу з використанням електронного квитка і систем відеоспостереження. Інтегрована технічна документація для придбання обладнання була розроблена в унісон для трьох заходів, і обладнання GPS було розроблено таким чином, щоб воно могло бути інтегровано з системою електронних квитків. Виклик тендера і відповідні документи були опубліковані в січні 2010 року. Тендерна процедура проходила через національну тендерну електронну систему, і переможець був номінований в березні 2010 року.

Монтажне і монтажне обладнання були виконані в рамках програми, створеної виробничою компанією транспортної компанії RAT. Система в основному встановлена і знаходиться на стадії тестування. На основних автостанціях встановлено 20 цифрових інформаційних панелей, а 80 автобусів оснащені бортовими комп'ютерами для відстеження GPS.

Крайова очікує, що цей захід буде:

- 1) підвищувати надійність системи громадського транспорту;
- 2) чи забезпечить кращий доступ до інформації про маршрут в режимі реального часу;
- 3) чи скоротить використання приватних автомобілів;
- 4) чи збільшить середню пропускну здатність на 10 відсотків;
- 5) чи збільшить зайнятість автотранспортних засобів на 5 відсотків.

Роттердам. Привабливість і популярність громадського транспорту

можна поліпшити, надаючи пасажиром надійну інформацію про сервіси в режимі реального часу.

Надання оновленої інформації про розклад через Інтернет або мобільний телефон стає все більш поширеним інструментом для компаній громадського транспорту. Таким чином, Роттердамська транспортна компанія RET вирішила ввести інформацію про прибуття / виїзди в режимі реального часу на 70 трамвайних, автобусних та метрових зупинках для зручності своїх пасажирів.

Динамічна інформація про громадський транспорт на високоякісній трамвайній лінії в Роттердамі почала функціонувати в жовтні 2002 року на 36 зупинках. RET придбала 60 нових трамваїв, які довелося адаптувати для установки в систему: спочатку вони не були виявлені системою, а це означало, що на інформаційних панелях були відображені неточні дані. Як тільки ця проблема була вирішена, RET провела успішні тести в період з жовтня по грудень 2002 року.

У жовтні 2003 року RET почав тестовий період для динамічної інформації на зупинках метро, і на початку 2004 року система була запущена з установкою 247 панелей на всіх 49 станціях метро міста. Панелі показують кількість хвилин, які пасажир повинні чекати до наступного напруги, а також інформацію про затримки.

До квітня 2004 року були встановлені перші 44 бортові системи зв'язку, і до січня 2005 року в систему було підключено в цілому 74 трамвая.

Незважаючи на високий рівень прийнятності, вплив заходів на кількість пасажирів не може бути визначено кількісно через падіння числа пасажирів під час здійснення, що частково пов'язано з впровадженням більш високих тарифів. Однак опитування, проведене в 2002 році, показало, що позитивна оцінка серед користувачів була високою. З 258 осіб, які відповіли, 47 відсотків погодилися з тим, що інформація про час очікування важлива, а 45 відсотків погодилися з тим, що динамічна інформація про громадський транспорт збільшує використання громадського транспорту.

1.4 Постановка завдання

Метою даного дипломного проекту є розробка макету інформаційного табло для міської зупинки громадського транспорту.

Об'єкт дипломної роботи – програмно-технічні засоби для виготовлення інформаційного табло на базі світлодіодного дисплею.

Предмет дипломної роботи – механізми міжмашиної взаємодії, як з точки зору концептуальної основи IoT.

Виходячи із сучасних тенденцій на сьогоднішній день (розширення функціоналу та зменшення габаритів) виникають наступні основні **вимоги** до пристрою, що розробляється:

- 1) забезпечення функціонування на базі контролера;
- 2) малі габарити;
- 3) передача отриманих даних з використанням бездротової технології.

Для досягнення поставленої мети слід вирішити такі **завдання**:

- 1) виконати огляд існуючих технологій моніторингу;
- 2) сформулювати необхідні вимоги щодо розроблюваного пристрою;
- 3) розробити алгоритм прогнозування часу прибуття транспортного засобу;
- 4) реалізувати веб-додаток для отримання прогнозу;
- 5) розробити макет пристрою відповідно до вимог;
- 6) виконати збірку пристрої;
- 7) виконати розрахунки по розділу «Охорона праці».

2. РЕАЛІЗАЦІЯ ПРОГРАМНИМИ ЗАСОБАМИ РОБОТИ ПРИСТРОЮ

Сучасна тенденція розвитку регіону призводить до збільшення кількості транспортних засобів у мережах дорожнього руху. До того ж саме рівень розвитку транспортної системи є однією з головних ознак економічної стабільності та процвітання регіону. Єдина транспортна система має забезпечувати розвиток і функціонування всіх видів транспорту, та максимально відповідати потребам регіону при мінімальних витратах. Все це, суттєво, сприяє економічному, соціальному і культурному розвитку суспільства.

Щодо збільшення ефективності транспортного обслуговування населення та організації дорожнього руху, то існує два шляхи: раціональне використання існуючої мережі та збільшення пропускної спроможності транспортної мережі. Перше рішення є достатньо матеріально витратним, тому що потребує реконструкцію транспортних вузлів і магістралей. Друге рішення пов'язане з раціоналізацією використання існуючих транспортних систем і оптимізацією процесів перерозподілу навантажень на транспортну мережу. Рішення такої технічної задачі повністю спирається на математичне моделювання процесів, що виникають при взаємодії існуючих транспортних мереж міст і їх споживачів. Тобто, основне завдання математичної моделі - спрогнозувати та оцінити ефективність можливих реалізацій проєктованих заходів.

В даний час істотну перевагу у цій сфері набувають інтелектуальні транспортні системи. Одним з найважливіших елементів сучасних інтелектуальних транспортних систем є застосування глобальних навігаційних супутникових систем, що в свою чергу, дає змогу вирішення цілої низки завдань аналізу таких систем, моделювання та прогнозування.

Серед цього ряду - завдання прогнозування часу руху транспортних засобів є одним з найбільш затребуваних і зрозумілих для кінцевого споживача,

як учасника дорожнього руху. Вирішення цього завдання необхідно не тільки для сповіщення пасажирів про час прибуття громадського транспорту на зупинки, але і для управління рухом і внесення своєчасних коригувань диспетчерськими службами.

Завдяки створенню відповідного апаратно-програмний комплекс «InTab» стане можливим автоматизувати процес моделювання процесів транспортних потоків.

Для цього необхідна формалізація параметрів, використовуваних аналітичних показників, які характеризують існуючий стан дорожньо-транспортної системи, це є першим етапом реалізації транспортної моделі міста, тобто створенням транспортної пропозиції. Складнішим і трудомістким, с точки зору, математичного моделювання завданням є наступний етап - розрахунок транспортного попиту.

Необхідно виділити наступні задачі, які потрібно вирішити під час створення інтелектуальної транспортної системи:

1. Організація безперервного автоматичного збору навігаційної інформації о місцезнаходженнях транспортних засобів.
2. Створення бази даних для збереження інформації про рух транспортних засобів.
3. Вибір, удосконалення та реалізація алгоритму, що дозволяє прогнозувати час руху транспортних засобів.
4. Інформування пасажирів про час прибуття громадського транспорту шляхом виводу повідомлення на інформаційне табло на зупинках.

За для вирішення поставлених задач, доцільно, скористатися багатофункціональною системою Wialon, яка застосовується для моніторингу рухомих і стаціонарних об'єктів на базі супутникових навігаційних систем ГЛОНАСС і GPS, стільникового зв'язку GSM (GPRS / SMS), супутникової зв'язку Inmarsat, Iridium, Thuraya і Globalstar; та для контролю руху встановити GPS-датчики до кожного транспортного засобу, які, в свою чергу, стануть

частиною тієї самої системи Wialon.

Все це, дає змогу відслідковувати та проаналізувати тимчасові та просторові координати транспортного засобу, що вже є базою для побудови алгоритму оцінки часу прибуття громадського транспорту на зупинки.

2.1 Бібліотека cURL

Процес авторизації і отримання інформації досить трудомісткий, тому було прийнято рішення використовувати бібліотеку cURL, яка використовується для перенесення даних з сервера або на сервер. За допомогою даної бібліотеки можна виконувати будь-які запити до API, що дозволить отримати доступ до даних і функцій Wialon Hosting в безперервному режимі та заповнити для зручності роботи власну базу даних.

PHP підтримує бібліотеку cURL, яка дозволяє підключатися і взаємодіяти з багатьма типами серверів з багатьма різними типами протоколів. В даний час підтримуються протоколи http, https, ftp, gopher, telnet, dict, file і ldap. Також підтримується HTTPS-сертифікати, HTTP POST, HTTP PUT, FTP-завантаження, завантаження на основі HTTP-форм, проксі-серверами, куки-файлами і аутентифікації користувача і пароля.

В роботі за допомогою бібліотеки cURL емулюватися відправка INPUT полів форми для авторизації в системі Wialon.

2.2 Протокол HTTP

HTTP (HyperText Transfer Protocol) - це протокол передачі гіпертексту

між розподіленими системами. По суті, http є фундаментальним елементом сучасного Web-програмування. HTTP забезпечує спілкування між безліччю хостів і клієнтів, а також підтримує цілий ряд параметрів мережі.

В основному, для спілкування використовується TCP/IP, але це не єдиний можливий варіант. За замовчуванням, TCP/IP використовує порт 80, але можна використовувати та інші.

Основне завдання протоколу - це забезпечення спілкування між безліччю веб-серверів і клієнтів. Спілкування між веб-сервером і клієнтом відбувається в два етапи: запит і відповідь. Клієнт відправляє HTTP запит на сервер, у відповідь на який отримує дані. Ініціатор з'єднання завжди є клієнт. Односпрямованість протоколу не дозволяє серверу посилати запити клієнтам (тільки відповідати на запити).

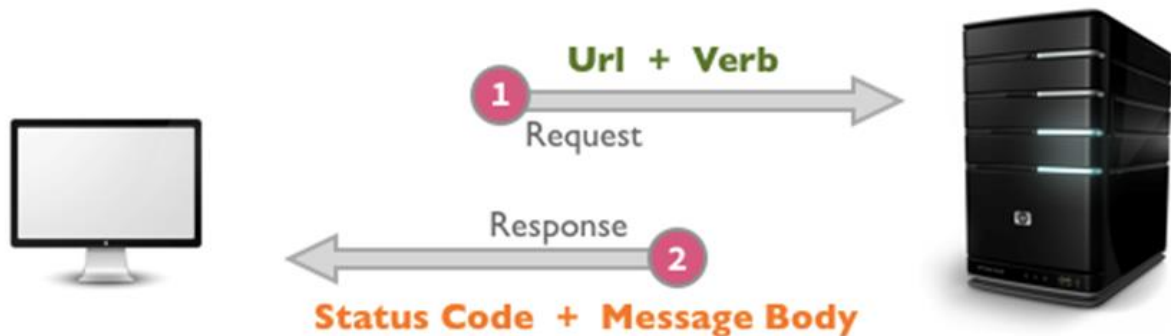


Рисунок 2.1 – Процес відправлення запиту клієнтом, обробка та відправка відповіді сервером

Кожен HTTP запит на веб-сервер не залежить від попередніх запитів. Протокол HTTP дозволяє вказати спосіб представлення (кодування) одного і того ж ресурсу за різними параметрами: mime-тип, мови і т. Д. Завдяки цій можливості клієнт і веб-сервер можуть обмінюватися двійковими даними, хоча даний протокол є текстовим.

Клієнт формує HTTP запит, який відправляється через Єдиний покажчик ресурсів (URL), у відповідь на який сервер дає відповідь (повідомлення). Структура URL дуже проста і складається з наступних компонентів:

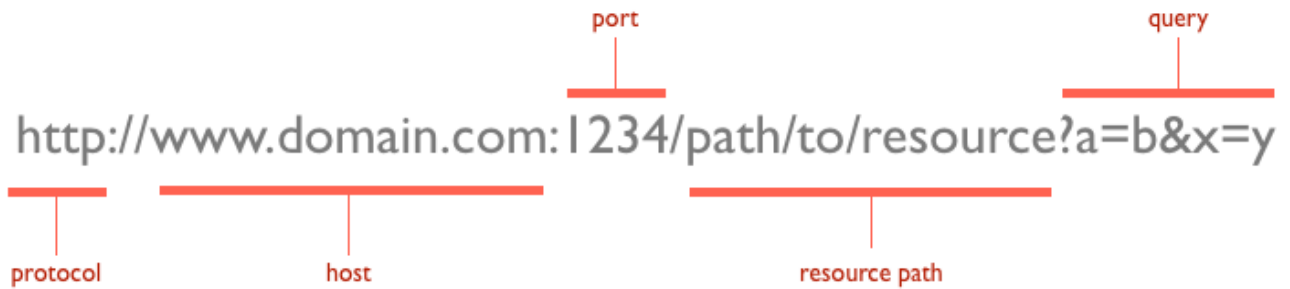


Рисунок 2.2 – Формат запиту

Протокол може бути, як HTTP для звичайних з'єднань, так і HTTPS для більш безпечного обміну даними. Далі слід шлях до ресурсу на сервері і ланцюжок параметрів. Оскільки в HTTP протоколі немає механізмів шифрування і механізмів безпеки, а базова аутентифікація по протоколу HTTP передає логін і пароль користувача в незашифрованому вигляді, то у HTTP протоколу з'явилося розширення HTTPS. HTTPS використовує TCP порт 433. Це розширення є зв'язкою двох протоколів: HTTP та SSL або HTTP і TLS.

Також HTTP може виступати в ролі транспорту для інших протоколів прикладного рівня, наприклад: SOAP, XML-RPC, JSON-RPC, WebDAV. Актуальна на даний момент версія протоколу, HTTP 1.1, описана в специфікації RFC 2616.

Структура HTTP протоколу дуже проста - це повідомлення плюс з'єднання. Повідомлення зазвичай складається з трьох частин, які передаються і клієнтами, і серверами в наступному порядку:

1. Стартовий рядок або рядок стану, яка визначає тип HTTP повідомлення. Стартовий рядок обов'язковий для будь-якого повідомлення.
2. Тема HTTP повідомлення, який може включати одне поле Host або кілька полів для передачі різної службової інформації.
3. Тіло HTTP повідомлення, яке містить HTTP об'єкти. Тіло повідомлення служить для передачі інформації користувача і є не у кожного HTTP повідомлення.

У HTTP протоколі є параметри, які виступають механізмом управління

взаємодії між клієнтом і сервером. І які визначають те, в якому вигляді інформація буде передана по HTTP протоколу:

1. Версія HTTP протоколу є його обов'язковим параметром і вказується в першому рядку будь-якого HTTP повідомлення.
2. URI - це параметр, який дозволяє однозначно ідентифікувати ресурс, який хоче запросити клієнт.
3. Дата та час – це також параметр HTTP протоколу.
4. HTTP протокол має параметри кодування повідомлень і кодування передачі.
5. Тип даних або медіа, тип - черговий параметр HTTP протоколу.
6. Лексеми програм дозволяють визначити розробника HTTP додатки і версію програми, лексема також є параметром HTTP протоколу.
7. HTTP протокол має параметри, які дозволяють визначити мову повідомлення, в HTTP це називається мовною міткою.
8. Іншою функцією в HTTP є одиниці вимірювання діапазонів, які дозволяють запросити тільки частина ресурсу.

HTTP є протоколом передачі даних сьомого рівня моделі OSI, що працює на основі технології клієнт-сервер, а також виступає в якості транспорту для інших протоколів, наприклад, так як JSON.

2.3 Формат обміну даними JSON

JSON (JavaScript Object Notation) - простий, заснований на тексті, спосіб зберігати і передавати структуровані дані, є форматом обміну даними, зручним для читання і написання як людиною, так і комп'ютером. Він заснований на підмножині мови програмування JavaScript, визначеного в стандарті ECMA-262 3rd Edition - December 1999. JSON - текстовий формат, повністю незалежний від

мови реалізації, але він використовує угоди, знайомі програмістам С-подібних мов, таких як С, С ++, С # , Java, JavaScript, Perl, Python і багатьох інших.

За допомогою простого синтаксису можна легко зберігати як прості цифри і рядки, так і масиви, об'єкти, використовуючи при цьому ніщо інше як текст. Так само можна пов'язувати об'єкти і масиви, що дозволяє створювати складні структури даних.

Після створення JSON рядка, його легко можна пересилати в будь-який додаток або комп'ютер, так як це всього лише текст.

JSON має багато переваг:

1. Компактність.
2. Легкий для прийняття людиною та машиною.
3. Легкість конвертації до програмних форматів: числові значення, рядки, булевої формат, нульове значення, масиви і асоціативні масиви.
4. Майже всі програмні мови мають функції, що дозволяють зчитувати і створювати json формат даних.

JSON засновано на двох структурах даних:

1. Колекція пар ключ / значення. У різних мовах, ця концепція реалізована як об'єкт, запис, структура, словник, хеш, іменованний список або асоціативний масив.
2. Впорядкований список значень. У більшості мов це реалізовано як масив, вектор, список або послідовність.

Це універсальні структури даних. Майже всі сучасні мови програмування підтримують їх в будь-якій формі. Логічно припустити, що формат даних, незалежний від мови програмування, повинен бути заснований на цих структурах.

Відносно задачі дипломного проекту, а саме збір навігаційної інформації о місцезнаходженнях транспортних засобів, необхідно реалізувати пошук по ідентифікатору.

Наприклад, потрібно дізнатися останнє розташування об'єкту з ID рівним

14157054. Для того, щоб у відповіді містилося розташування об'єкта потрібно встановити прапор 0x00000400 (1024).

Формат запиту:

```
https://hst-
api.wialon.com/wialon/ajax.html?svc=core/search_item&params={"id":
14157054,"flags":1024}&sid=06cea7c52d75c02e98e2eeb012b684ce
```

Результат виконання:

```
{"item":{"pos":{"t":1515693202,"f":1073741827,"lc":0,"y":48.9358699,"x":3
8.5215632,"z":100,"s":6,"c":182,"sc":16},"lmsg":{"t":1515693202,"f":1073741827,
"tp":"ud","pos":{"y":48.9358699,"x":38.5215632,"z":100,"s":6,"c":182,"sc":16},"i":
4,"lc":0,"p":{"gsm":2,"param202":0,"param204":0,"param240":0,"pwr_ext":27.188
,"pwr_int":3.862,"param201":0,"param203":0,"battery_charge":0}}},"flags":1024}
```

З відповіді видно, що об'єкт із заданим ідентифікатором має значення координат (48.9358699, 38.5215632). Однак для перетворення рядка JSON в змінну PHP використовується метод `json_decode()`.

Нижчих наведено фрагмент коду, Який реалізує такий запит та представляє результат виконання у потрібний формат для бази даних за допомоги json-інтерпретатору.

```
$result = $wialon_api->login($token);
$json = json_decode($result, true);
if(!isset($json['error']))
{
$result=$wialon_api-core_search_item({'id':14157054,'flags':1024});
$jn_result = json_decode($result, true);
```

```

$jn_x = $jn_result['item']['pos']['x'];
$jn_y = $jn_result['item']['pos']['y'];
$jn_s = $jn_result['item']['pos']['s'];
$time = date('H:i:s');
$result = mysqli_query ($db, "INSERT INTO tr (id,x,y,s,time)
VALUES('14157051','$jn_x','$jn_y','$jn_s','$time')");
$wialon_api->logout();
}
else echo WialonError::error($json['error']);

```

2.4 Синхронізація часу за рахунок NTP серверу

З часу початку використання комп'ютерів (спершу в спеціалізованих системах, в потім і в повсякденному житті) з'явилося завдання підтримувати точність ходу системного годинника, тобто мати точно синхронізований час з іншими комп'ютерами і системами. Особливо точність часу критична в розподілених системах, які критичні до порядку обчислення завдань, обробки даних і т.п.

Одним із прикладів такої синхронізації, де вперше був застосований популярний протокол синхронізації часу NTP, може бути диспетчерська служба управління транспортними засобами.

Також важливо, щоб сама мережа комп'ютерів мала не тільки синхронізований час, але і цей час був точним з іншими системами по всій планеті. Тобто час по всій території планети був однаковим (або досить близьким).

2.4.1 Протоколи DAYTIME і TIME

Першими протоколами точного часу, використовуваним на комп'ютерах, були DAYTIME (RFC 867) і TIME (RFC 868). Перший призначався для повідомлення дати і часу в зрозумілому людині вигляді, другий - зрозумілому комп'ютера вигляді. Формат відповіді DAYTIME суворо не регламентується і не призначений для машинної обробки - передбачається лише, що людині, прочитав отриманий рядок, стане ясно поточний час.

Протокол TIME, навпроти, призначений для обміну часу між машинами. На підключився до TIME-сервера комп'ютер приходить UDP-пакет, що містить єдине 32-бітове беззнакове число, відповідне числу минулих з 1 січня 1900 р секунд по UTC. Оскільки таке число переповнюється через 136 років, цей протокол здатний функціонувати тільки до 2036 р.

2.4.2 Протокол NTP

Зрозуміло, що ні DAYTIME, ні TIME не можуть забезпечити необхідну точність синхронізації часу. У зв'язку з цим, в 1985 р Девідом Л. Міллсом (David L. Mills) з університету Делавера був розроблений мережевий протокол синхронізації часу NTP, точніше його початкова, пізніше названа нульовою (NTPv0) версія, описана в RFC 958. Протокол NTP використовує алгоритм Марзулло (запропонований Кейтом Марзулло (Keith Marzullo) з Університету Каліфорнії, Сан-Дієго), включаючи таку особливість, як облік часу передачі. У версії 4 здатний досягати точності 10 мс при роботі через Інтернет, і до 0,2 мс всередині локальних мереж. NTP для синхронізації використовує протокол UDP і 123 порт, DAYTIME - 13 порт TCP / UDP, TIME - 37 порт TCP / UDP.

Протокол NTP удосконалювався не один раз: NTPv1 (1988 р, RFC 1059), NTPv2 (1989 р RFC1119), NTPv3 (1992 р RFC1305), NTPv4 (1996 р RFC2030).

Для визначення точності або значущості того чи іншого NTP-сервера використовують параметр Stratum (стратум) - ціле число від 1 до 15. Стратум 1 відповідає серверам, які мають безпосередньо зв'язок з еталонному часу, стратум 2 - сервер, який отримує відомості про час від серверів першого стратума і т.д. При побудові ланцюжка зв'язків значення стратума збільшується на 1.

Робота алгоритму NTP досить проста і може бути продемонстрована таким чином, по чотирьом даними: час відправлення запиту (по годиннику клієнта); час отримання запиту сервером (по годиннику сервера); час відправлення відповіді сервером (по годиннику сервера); час отримання відповіді (по годиннику клієнта) можна знайти час пакета в дорозі туди і назад, а потім - відкоригувати локальний час.

При цих розрахунках ми користуємося трьома важливими припущеннями:

1. Пакет проходить шлях від клієнту до сервера і назад за рівний час.
2. Швидкість ходу годинника клієнта і сервера дорівнює.
3. На обчислення нового локального часу не йде додатковий час.

Насправді всі ці припущення, строго кажучи, не вірні, і отримати точне значення серверного часу за допомогою одного NTP-запиту неможливо. Тому для синхронізації годин зазвичай використовується кілька NTP-серверів, на які постійно шлються запити. Накопичуючи статистику за тривалий час, математичними методами можна визначити точність показань кожного з серверів, швидкість ходу годинника на кожному з них, і т.п. величини, використовуючи які, можна домогтися математично доказової точності синхронізації. Конкретні використовувані методи описані в RFC і надзвичайно складні. До речі, через третє припущення, використання синхронізації по NTP по несиметричним каналам зв'язку (супутникові тощо) не правомірно.

Крім власне обміну показаннями часу, в NTP починаючи з версії 2 включений механізм обміну метаінформацією у вигляді "керуючих повідомлень" NTP. Формат самих керуючих повідомлень не заданий в RFC, але існуючий стандарт де-факто дозволяє з їх допомогою запитувати у NTP-сервера такі параметри, як адреси всіх його клієнтів і вищих серверів, затримку до кожного з них і т.п.

NTP дозволяє домогтися високоточної синхронізації часу в мережі синхронізуються серверів, кожен з яких отримує свідчення з декількох джерел, обробляє їх, і передає далі. Він застосовується лише всередині невеликих локальних мереж і мереж з малими затримками пакетів; в Інтернеті він практично непридатний через велику (і, що важливо, випадкову) затримку пакетів, яка на порядки перевершує різницю в показаннях годинника клієнта і сервера. Така NTP-мережа характеризується масштабністю і стійкістю до збоїв - навіть в разі відмови годин одного з серверів інші негайно це помітять і перестануть використовувати його свідчення.

2.4.3 Протокол SNTP

Крім NTP, існує спрощена версія цього протоколу - SNTP (Simple Network Time Protocol). Він реалізований для синхронізації часу кінцевим клієнтом, оскільки всі переваги протоколу NTP проявляються саме в мережі серверів, а для отримання показань кінцевим користувачем NTP зайво складний. Тому для синхронізації часу кінцевими комп'ютерами і серверами був запропонований протокол SNTP (SNTPv3: 1992 р RFC1361 і 1995 року, RFC1769; SNTPv4 включений як підпротокол в NTPv4). Насправді SNTP - це не новий протокол, а спосіб використання NTP-пакетів і NTP-серверів в додатках, де не потрібно високоточне час, або воно недосяжне. У цьому

випадку клієнт використовує тільки частину інформації UDP-пакета NTP-сервера. SNTP-клієнт може працювати з будь-якими версіями NTP-серверів, і крім них - з особливими SNTP-серверами, які у відгуках заповнюють тільки необхідні дані UDP-пакета. Таким чином, "полегшений" SNTP утворює не мережа синхронізуються серверів, а пари "клієнт-сервер". Будь-який NTP-сервер є одночасно SNTP-сервером. Клієнт, що не передає отримане час далі, може працювати як NTP- або SNTP-клієнт, в залежності від умов. Для SNTP, як і для NTP, зарезервовані 123-ї UDP-порт.

2.5 Протокол для взаємодії Інтернет Речей

Наступним етапом роботи є використання протоки MQTT.

MQTT або Message Queue Telemetry Transport - це легкий, компактний і відкритий протокол обміну даними створений для передачі даних на віддалених локаціях, де потрібно невеликий розмір коду і є обмеження по пропускну здатності каналу. Перераховані вище гідності дозволяють застосовувати його в системах M2M (Машинно-Машинне взаємодія) і IIoT (Промисловий Інтернет речей). Також існує версія протоколу MQTT-SN (MQTT for Sensor Networks), раніше відома як MQTT-S, яка призначена для вбудованих бездротових пристроїв без підтримки TCP / IP мереж, наприклад, Zigbee.

Переваги:

1. Простий у використанні. Протокол являє собою програмний блок без зайвої функціональності, який може бути легко вбудований в будь-яку складну систему.
2. Шаблон проектування видавець-передплатник зручний для більшості рішень з датчиками. Дає можливість пристроям виходити на зв'язок і публікувати повідомлення, які не були заздалегідь відомі або визначені.

3. Легкий в адмініструванні.
4. Знижено навантаження на канал зв'язку.
5. Робота в умовах постійної втрати зв'язку або інших проблем на лінії.
6. Немає обмежень на формат переданих даних.

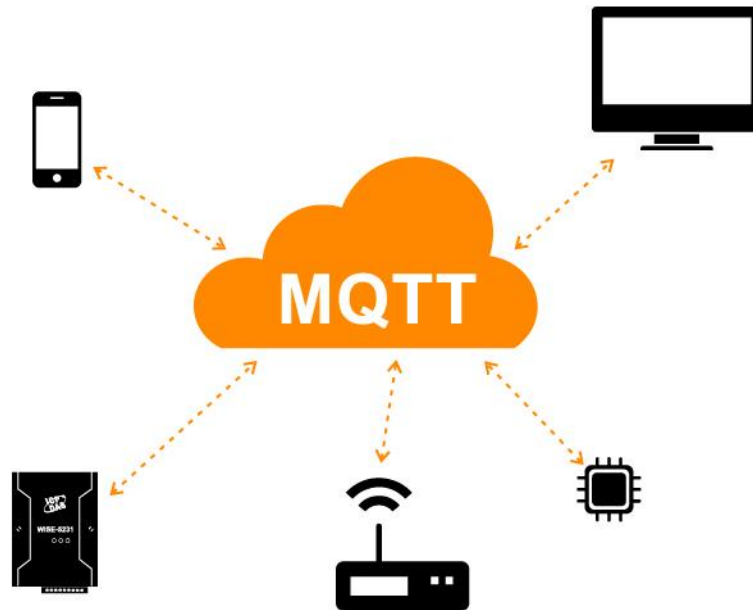


Рисунок 2.3 – Протокол обміну даними

Основні особливості протоколу MQTT:

1. Асинхронний протокол.
2. Компактні повідомлення.
3. Робота в умовах нестабільного зв'язку на лінії передачі даних.
4. Підтримка декількох рівнів якості обслуговування (QoS).
5. Легка інтеграція нових пристроїв.

Протокол MQTT працює на прикладному рівні поверх TCP / IP і використовує спочатку 1883 порт (8883 при підключенні через SSL).

Обмін повідомленнями в протоколі MQTT здійснюється між клієнтом (client), який може бути видавцем або передплатником (publisher / subscriber) повідомлень, і брокером (broker) повідомлень (наприклад, Mosquitto MQTT).

Видавець відправляє дані на MQTT брокер, вказуючи в повідомленні певну тему, топік (topic). Передплатники можуть отримувати різні дані від безлічі видавців залежно від підписки на відповідні топіки.

Пристрої MQTT використовують певні типи повідомлень для взаємодії з брокером, нижче представлені основні:

1. Connect - встановити з'єднання з брокером.
2. Disconnect - розірвати з'єднання з брокером.
3. Publish - опублікувати дані в топік брокера.
4. Subscribe - підписатися на топік брокера.
5. Unsubscribe - відписатися від топіка.



Рисунок 2.4 – Схема простого взаємодії між передплатником, видавцем і брокером

MQTT повідомлення складається з декількох частин:

1. Фіксований заголовок (мається на всіх повідомленнях).
2. Змінний заголовок (присутній тільки в певних повідомленнях).
3. Дані, «навантаження» (присутній тільки в певних повідомленнях).

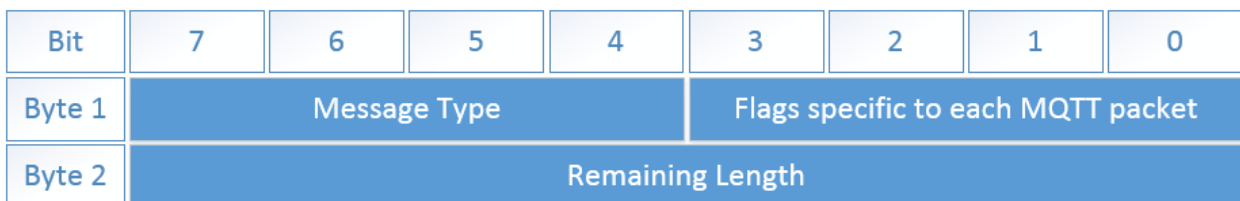


Рисунок 2.5 – структура повідомлень

Message Type - це тип повідомлення, наприклад: CONNECT, SUBSCRIBE, PUBLISH і інші.

Flags specific to each MQTT packet - ці 4 біта відведені під допоміжні прапори, наявність і стан яких залежить від типу повідомлення.

Remaining Length - представляє довжину поточного повідомлення (змінний заголовок + дані), може займати від 1 до 4 байта.

Всього в протоколі MQTT існує 15 типів повідомлень:

**K – клієнт, **C – сервер*

Таблиця 2.1 – Типи повідомлень протоколу MQTT

Тип сообщения	Значение	Направление передачи	Описание
Reserved	0000 (0)	нет	Зарезервирован
CONNECT	0001 (1)	K* -> C**	Запрос клиента на подключение к серверу
CONNACK	0010 (2)	K <- C	Подтверждение успешного подключения
PUBLISH	0011 (3)	K <- C, K -> C	Публикация сообщения
PUBACK	0100 (4)	K <- C, K -> C	Подтверждение публикации
PUBREC	0101 (5)	K <- C, K -> C	Публикация получена
PUBREL	0110 (6)	K <- C, K -> C	Разрешение на удаление сообщения
PUBCOMP	0111 (7)	K <- C, K -> C	Публикация завершена
SUBSCRIBE	1000 (8)	K -> C	Запрос на подписку
SUBACK	1001 (9)	K <- C	Запрос на подписку принят
UNSUBSCRIBE	1010 (10)	K -> C	Запрос на отписку
UNSUBACK	1011 (11)	K <- C	Запрос на отписку принят
PINGREQ	1100 (12)	K -> C	PING запрос
PINGRESP	1101 (13)	K <- C	PING ответ
DISCONNECT	1110 (14)	K -> C	Сообщение об отключении от сервера
Reserved	1111 (15)		Зарезервирован

Для побудови систем промислового Інтернету речей, а також інтегрування існуючого обладнання в хмарні системи використовуються спеціальні пристрої.

1. Web-програмовані PoT контролери.
2. PoT модулі віддаленого вводу / виводу.
3. Шлюзи PoT.

4. IoT платформи, сервера MQTT.
5. Вимірники CO, CO₂, температури і вологості.



Рисунок 2.5 – Устаткування для IoT з підтримкою MQTT

3 РОЗРОБКА ІНФОРМАЦІЙНОГО ТАБЛО НА БАЗІ СВІТЛОДІОДНОГО ДИСПЛЕЮ

Для виготовлення інформаційного табло на базі світлодіодного дисплея знадобляться: контролер, світлодіодні модулі, блоки живлення й проводи для підключення, профілю та куточки з алюмінію, а також кріпильні металовироби та герметик.

У такій схемі LED-модулі несуть безпосередню інформацію, тобто формують літери в рухомому рядку. Вони бувають одно- або різнокольоровий.

Розмір профілю, що утворює корпус для світлодіодного дисплея, відрізняється в залежності від місця установки табло.

3.1 Модулі для зборки

Екран потрібних габаритів збирається з готових електронних блоків (модулів) стандартних розмірів, укомплектованих пікселями з світлодіодів або зборок RGB, з'єднаними на загальній платі та мають необхідні роз'єми і шлейфи для об'єднання з сусідніми блоками.

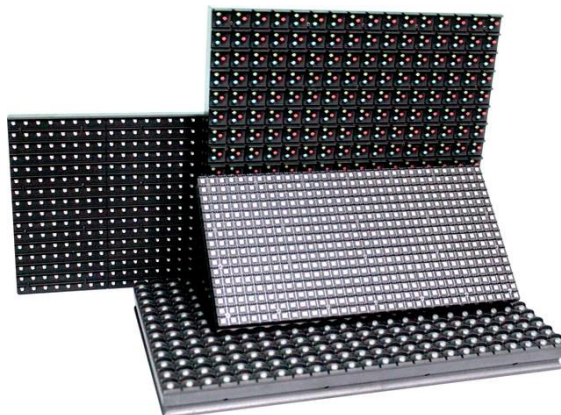


Рисунок 3.1 – Світлодіодні модулі

Набор типових параметрів мають модулі P10:

- 1) розмір, мм - 320 x 160 x 20;
- 2) вага модуля, г - 600-700;
- 3) крок пікселя, мм - 10;
- 2) роздільна здатність (кількість пікселів на 1 м²) - не менше 256 x 192;
- 3) яскравість світлодіодного екрана, кд / м² - 6 000-7 000;
- 4) кут половинній яскравості, градус - 120;
- 5) термін служби, годин - до 50 000;
- 6) максимальна споживана потужність (для вуличних екранів), Вт / м² - 500;
- 7) відстань комфортної видимості зображень, м - від 7;
- 8) всі світлові та електронні компоненти захищені від впливу вологи, пилу, механічних впливів.

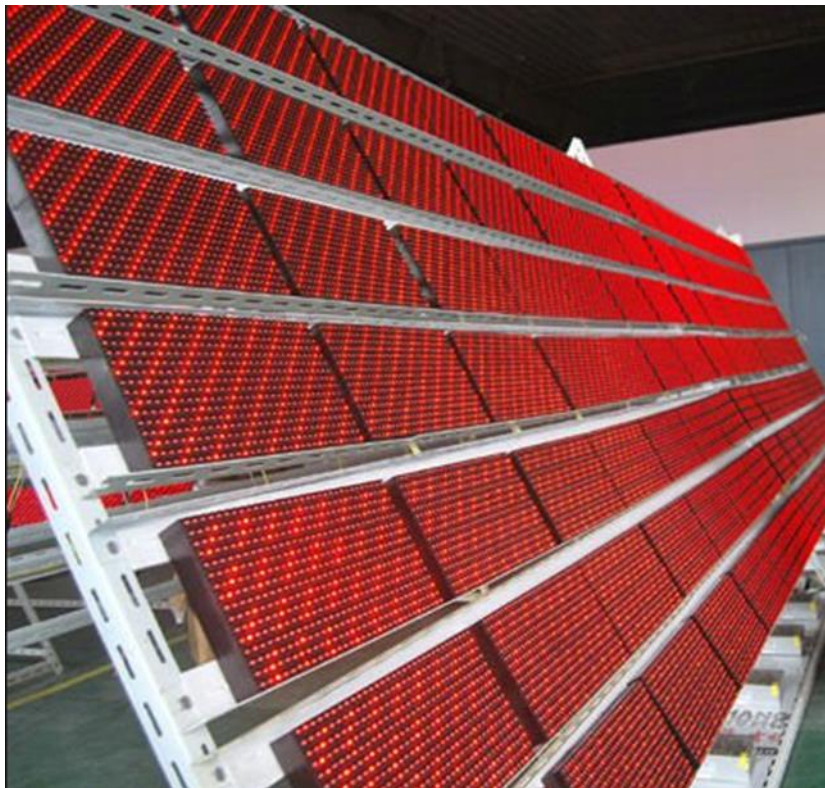


Рисунок 3.2 – Типовий варіант каркасу

На першому етапі виготовлення інформаційного табло необхідно виготовити надійну несучу металоконструкцію для розміщення на ній великої кількості електронних блоків (модулів, контролерів, джерел живлення - драйверів, що перетворюють мережеву змінну напругу 220 В в постійну - 12 В). Конструкція являє собою каркас із квадратної профільної труби.

На другому етапі збирають модулі Р10, кріплять до каркаса впритул один до одного та з'єднують за допомогою шлейфів, що мають якісні роз'єми «тато-мама». Кріплення модулів часто здійснюється за допомогою надійних магнітів, що дуже спрощує стадію збірки та особливо розбирання при виконанні ремонтних робіт.

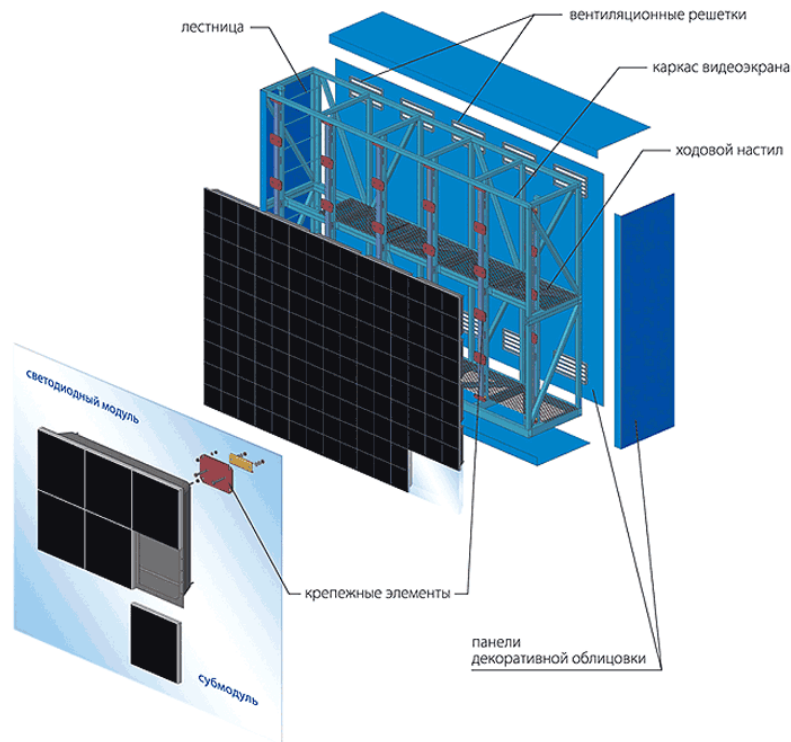


Рисунок 3.3 – Схема побудови світлодіодного дисплея

Далі із зворотного боку каркаса розміщуються блоки живлення та контролери, які відповідають за обробку відеоінформації і розподіл її на конкретні модулі та малі пікселі. Задня стінка виготовляється з металевого листа або алюмінієвої композитної панелі.

3.2 Модулі для зборки макету

Для розробки макету були використані світлодіодні модулі MAX7219, такий вибір було зроблену через невелику ціну таких модулів та схожість в роботі з модулем P10. Модулі MAX7219 постачають по чотири штуки в одній лінійці.



Рисунок 3.4 – Світлодіодні модулі MAX7219

Набір типових параметрів мають модулі MAX7219:

- 1) розмір, мм - 32 x 32 x 15;
- 2) частота тактування інтерфейсу SPI, МГц, -10;
- 3) напруга живлення, В, - 4.4...5;
- 4) споживання по шині +5 В в сплячому режимі, мкА , - 150;
- 5) струм через один сегмент в імпульсі, мА , - до 300;
- 6) середній струм через один сегмент, мА , - до 40.

Модулі підключаються послідовно, для збору табло необхідних габаритів використовуються модулі стандартного розміру.

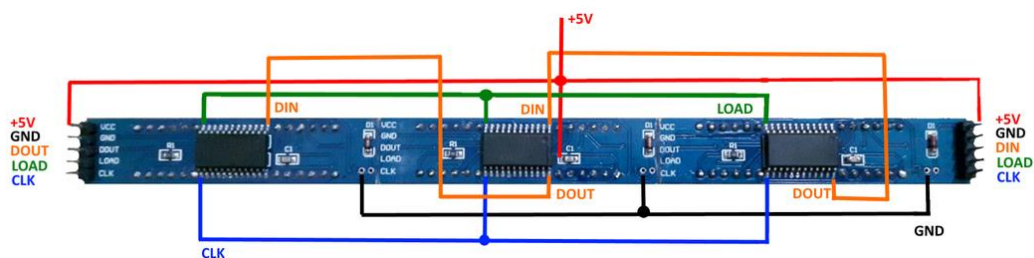


Рисунок 3.5 – Послідовне підключення модулів

3.3 Блок живлення

Для живлення табло на базі модулів P10 необхідний блок живлення 5В 40А 200Вт. Для нормальної роботи модулів MAX7219 необхідний блок живлення 3.3 В 28А.

Особливості:

1. Вхідна напруга вибирається перемикачем.
2. Захист від короткого замикання, перевантаження, високої напруги, високої температури.
3. Охолодження природною конвекцією.
4. Витримує вхідний перенапруження 300vac протягом 5 секунд.
5. Вбудований обмежувач постійного струму.
6. Протестовано при 100% навантаженні.
7. LED індикатор живлення.
8. Фіксована частота перемикачання 90 КHz.

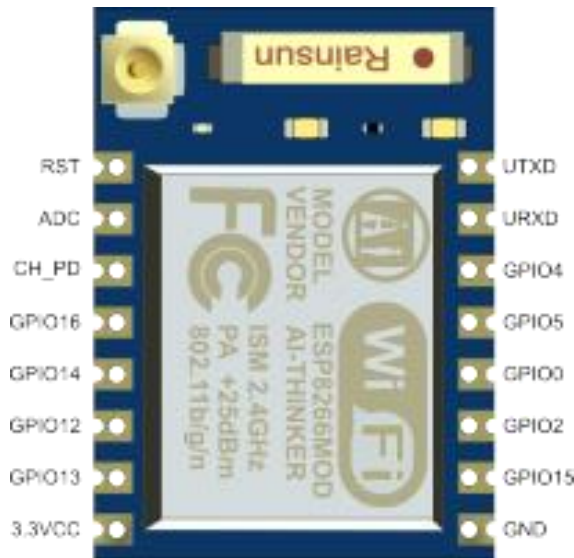


Рисунок 3.6 – Блок живлення 5В 40А 200Вт

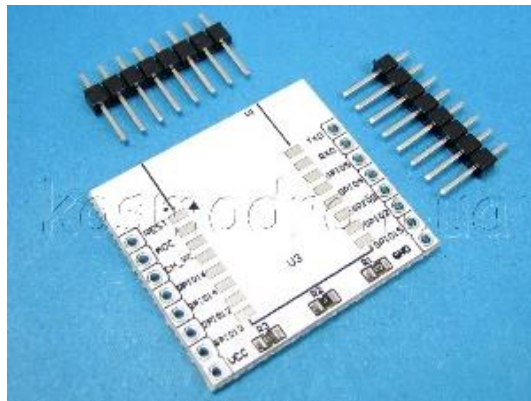
3.4 Контролер

ESP8266-07 представляє собою WiFi модуль з ультра низьким

споживанням. Сам чіп проектувався для пристроїв зі світу інтернет речей, дана плата не дозволяє розробнику одразу підключити контролер до комп'ютера через відсутність вмонтованого USB- UART інтерфейсу. Також для нормальної роботи необхідно через спеціальний адаптер перейти від кроку гребінки 2 мм до 2,54 мм.



а)



б)

Рисунок 3.7 – WI-FI модуль ESP8266-07(а) та адаптер для ESP8266-07(б)

Після переходу на новий крок гребінки, для нормальної роботи WI-FI модуля необхідно зібрати наступну схему з підтримкою завантаження скетчу з Arduino IDE, UDK, Sming та без неї.

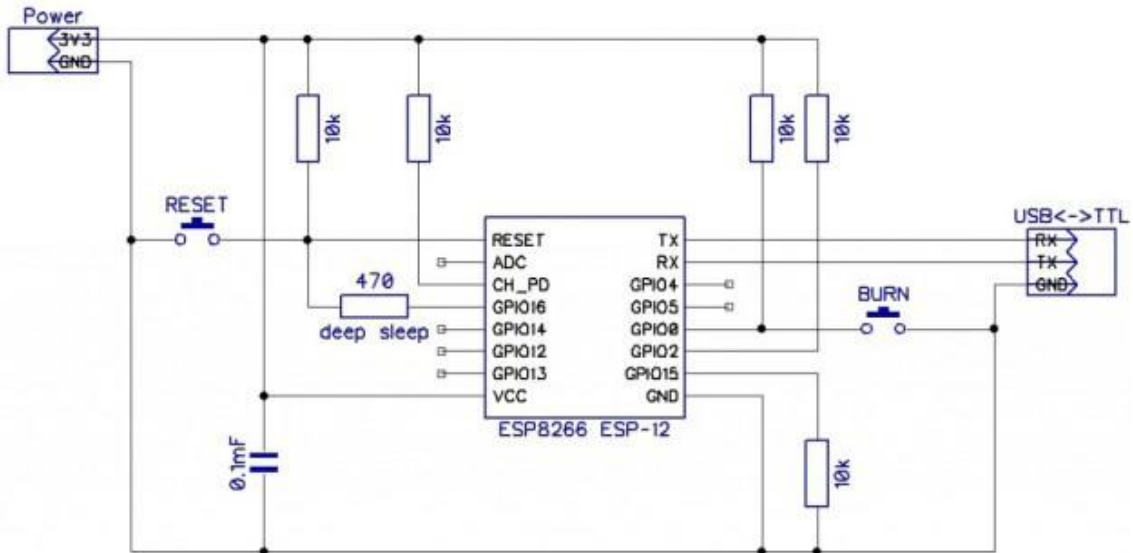


Рисунок 3.8 - Схема підключення ESP8266 (без автозавантаження прошивки, прошиваються попередньо встановивши перемичку BURN і перезавантаживши модуль)

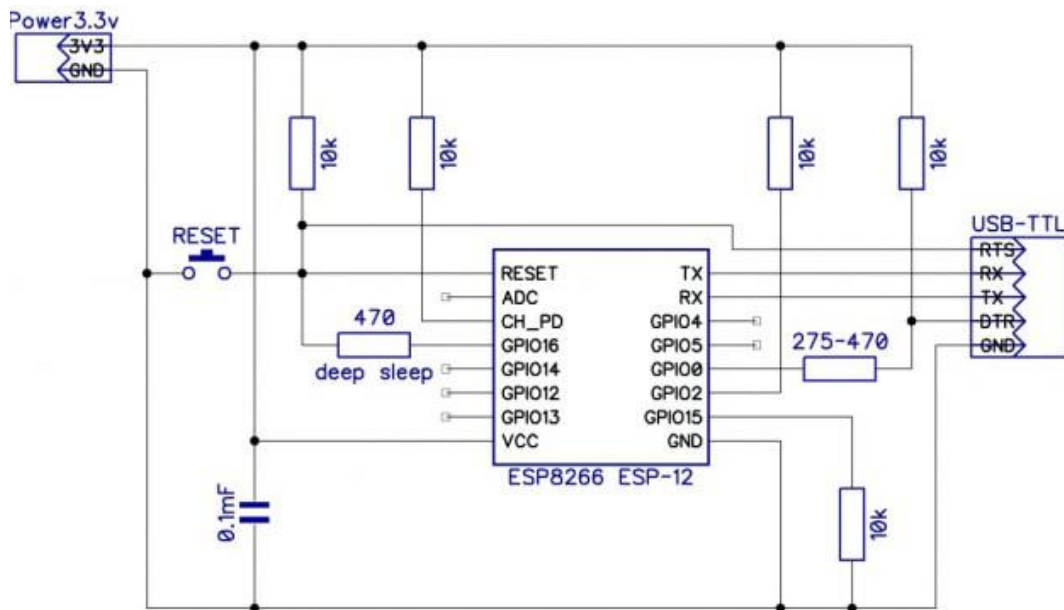


Рисунок 3.9 - Схема підключення з підтримкою автозавантаження прошивки з Arduino IDE, UDK, Sming.

Наступний крок це підключення вже укомплектованого модулю до комп'ютера, це робиться через USB інтерфейс за допомогою USB-TTL конвертеру. Треба звернути увагу, що виходи TX, RX на ESP8266 переплутані місцями.



Рисунок 3.10 - USB-TTL конвертер PL2303HX USB-UART

Для управління інформаційним табло необхідно написати скетч за допомогою програмного забезпечення ArduinoIDE, потім залити його у вбудовану пам'ять мікроконтролера.

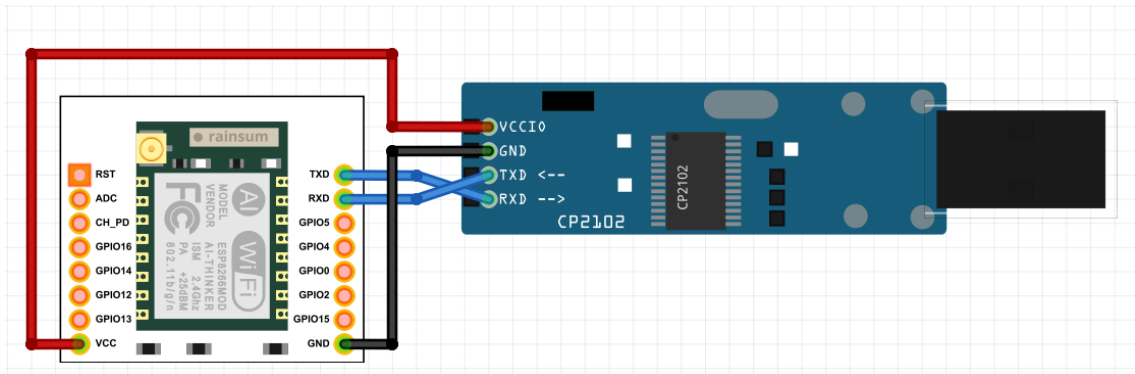


Рисунок 3.11 - Схема підключення ESP2866-07 до USB-TTL

Після завантаження скетчу до контролера ,необхідно підключити світлодіодні модулі , та елементи живлення.

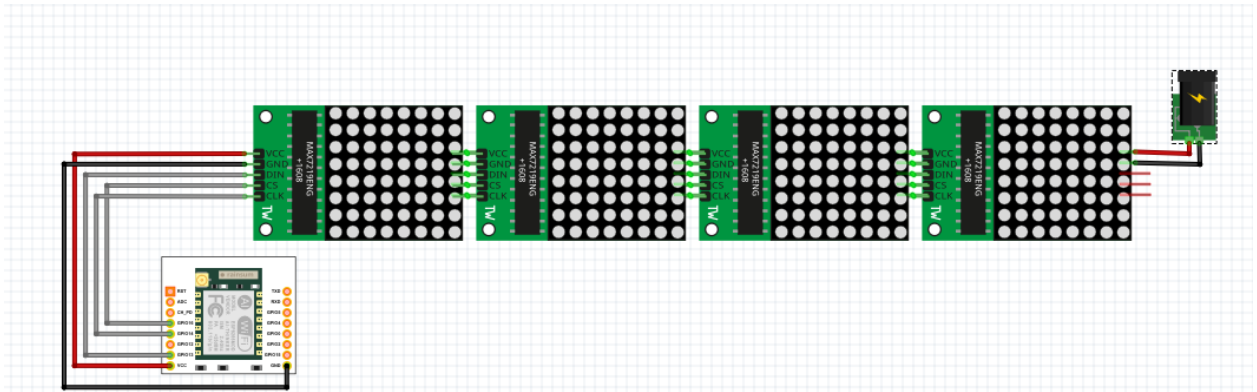


Рисунок 3.12 - Схема підключення компонентів інформаційного табло

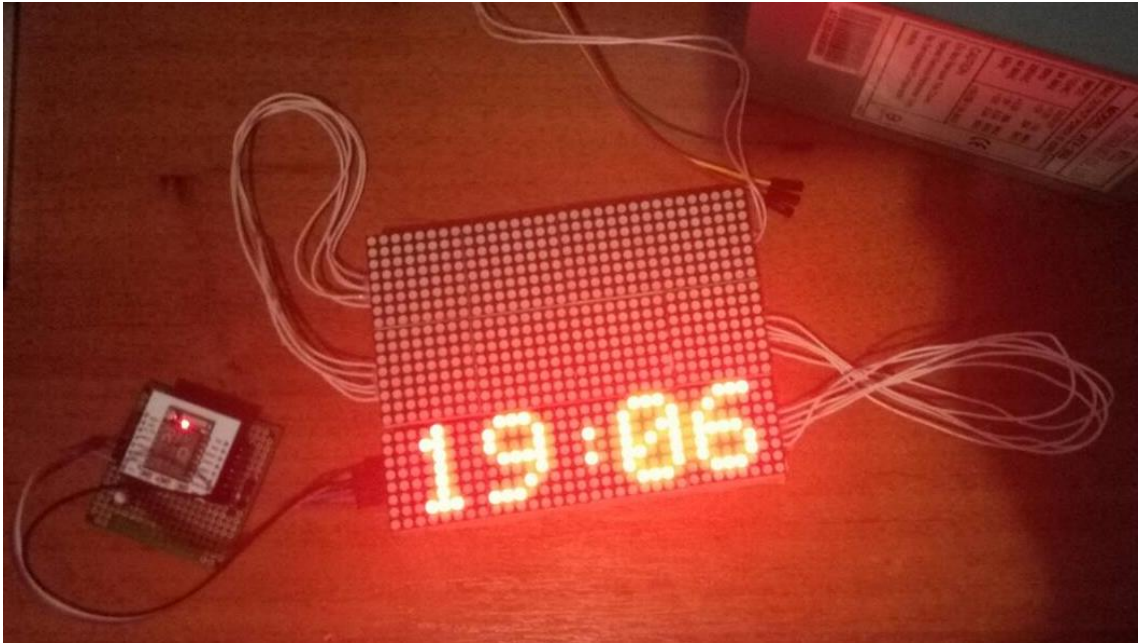


Рисунок 3.13 – Вигляд інформаційного табло без корпусу

3.4 Архітектура системи

В якості системи спостереження GPS стосовно транспортних засобів використовується багатофункціональна система для моніторингу рухомих і стаціонарних об'єктів на базі супутникових навігаційних систем ГЛОНАСС і GPS, стільникового зв'язку GSM (GPRS / SMS), супутникового зв'язку Inmarsat, Iridium, Thuraya і Globalstar - система Wialon.

Попередньо, для розгортання системи прогнозування в транспортних засобах встановлені термінали на рис. 3.14.



Рисунок 3.14 – Загальний вигляд терміналу

Функціонально термінал складається з мікроконтролера, незалежної пам'яті, ГЛОНАСС / GPS-модуля, GSM-модуля, цифрового інтерфейсу RS485, аналогових (дискретних) входів, імпульсних (дискретних) входів, дискретного виходу, акселерометра, інтерфейсу карти пам'яті (опціонально).

Дані, що передаються терміналом:

- час і дата за Гринвічем;
- координати (широта, довгота, висота);
- швидкість, прискорення і напрям руху;
- кількість супутників в рішенні навігаційного завдання;
- фактор втрати точності в горизонтальній площині;
- значення напруги на аналогових входах;
- значення з імпульсних входів;
- інформація про події, що відбулися;
- дані з датчиків рівня палива по інтерфейсу RS458;
- стан дискретних виходів;
- статус пристрою.

Так само термінал дозволяє здійснювати управління зовнішнім устаткуванням за допомогою дискретного виходу, використовуючи команди по GPRS або SMS. Накопичені дані передаються через мережу оператора стільникового зв'язку стандарту GSM 850/900/1800/1900 за допомогою технології пакетної передачі даних GPRS на виділений сервер зі статичною IP-адресою або доменним ім'ям, з якого можуть бути отримані через мережу Інтернет для подальшого аналізу і обробки на пультах диспетчерів. Повністю зарядженої акумуляторної батареї становить 4 години на повнофункціональному режимі і додатково 3 години в режимі збереження даних, але може змінюватися в залежності від стану GSM-зв'язку. На рисунку 3.15 описується алгоритм роботи системи прогнозування, що включає опитування датчиків, власне збір та аналіз даних для отримання прогнозу часу прибуття транспортного засобу на пункти зупинки і відображення отриманої

інформації на інформаційне табло.

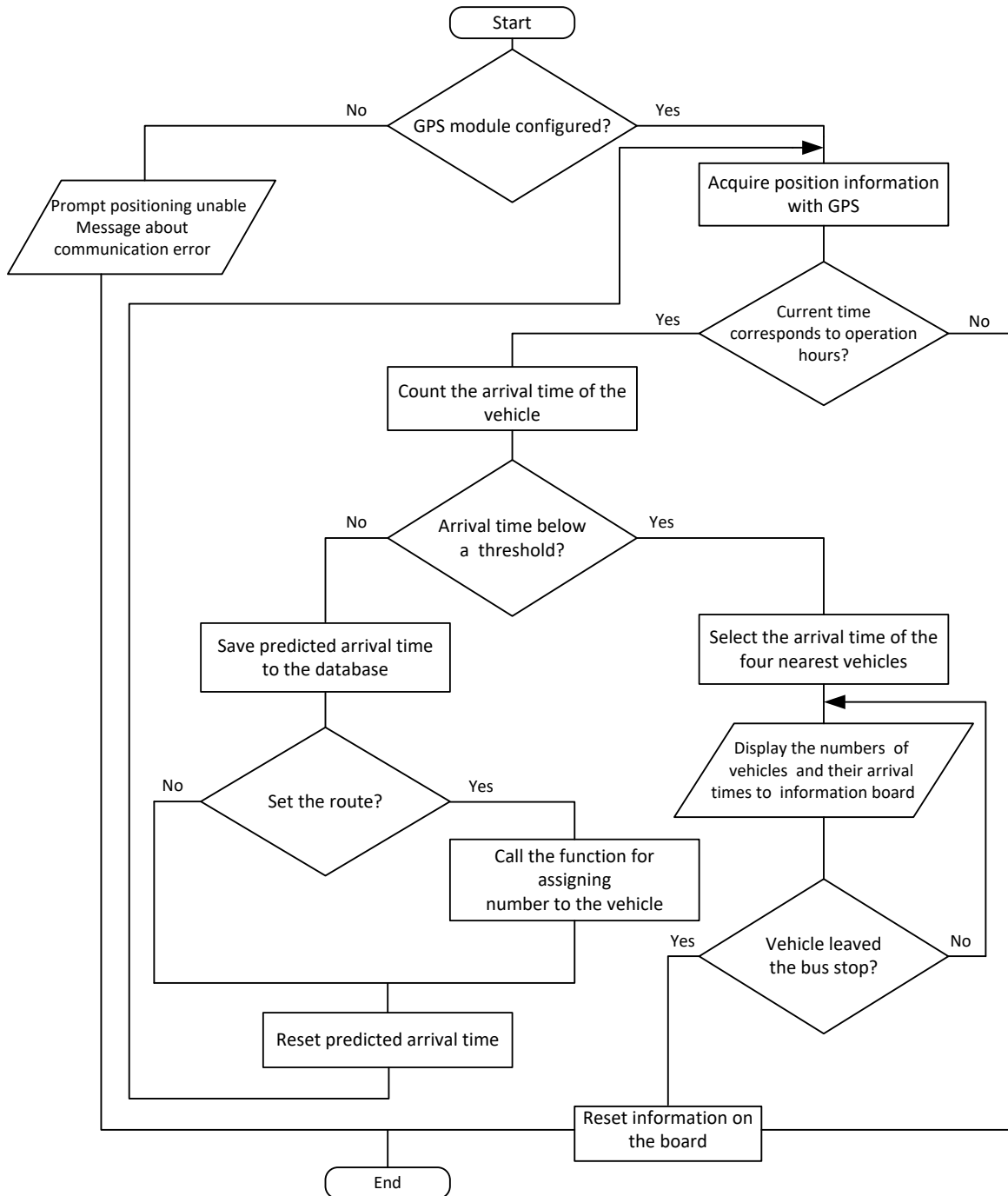


Рисунок 3.15 – Алгоритм роботи системи прогнозування

Система прогнозування являє собою web-додаток, яке розміщується на виділеному сервері, що має власне доменне ім'я, з якого можна, за допомогою запиту через мережу Інтернет, отримати для подальшого виведення час прибуття транспортного засобу на інформаційне табло.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даному розділі проведено аналіз потенційних небезпечних причин. Розглянуті заходи, які дозволяють забезпечити гігієну праці та виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики. Так як завданням дипломної роботи є розробка не тільки макету інформаційного табло на базі світлодіодного дисплею, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для серверу, де буде знаходитися веб-додаток програмно-апаратного комплексу «InTab».

4.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Основними характеристиками персонального комп'ютера є наступні:

- 1) робоча напруга $U=+220\text{В} \pm 5\%$;
- 2) робочий струм $I=2\text{А}$;
- 3) споживана потужність $P=350\text{ Вт}$.

Роботу користувача розробленої підсистеми слід віднести до категорії Ia (легкі фізичні роботи) відповідно до даної категорії відносяться всі види діяльності, які виконуються сидячи й не вимагають фізичного напруження.

При експлуатації даного програмного продукту відповідно існують наступні небезпечні й шкідливі виробничі фактори:

1. фізичні:
 - а. підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини;

- b. підвищена або знижена вологість повітря;
 - c. підвищений рівень статичної електрики;
 - d. підвищена напруженість електричного й магнітного полів;
 - e. відсутність або нестача природного світла;
 - f. знижена освітленість робочої зони;
 - g. підвищений рівень шуму на робочому місці;
 - h. підвищений рівень електромагнітного випромінювання;
 - i. знижена контрастність.
2. психофізіологічні:
- a. фізичні перевантаження: статичні та динамічні;
 - b. нервово-психічні перевантаження: розумове перенапруження, монотонність праці, перенапруження аналізаторів та емоційні перевантаження.

4.2 Електробезпека

Основним небезпечним фактором при роботі з ЕОМ є небезпека ураження людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані виявити наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм справляє на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (подразнення і збудження нервових волокон та інших органів тканин організму) дій.

- 1) Тяжкість ураження людини електричним струмом залежить від цілого ряду чинників: значення сили струму;
- 2) електричного опору тіла людини і тривалості протікання через

нього струму;

- 3) роду і частоти струму;
- 4) індивідуальних властивостей людини і навколишнього середовища.

Відповідно до, приміщення для ЕОМ відноситься до приміщень без підвищеної небезпеки, тобто до приміщень, у яких відсутні умови, що створюють підвищену або особливу небезпеку. Небезпека ураження електричним струмом існує всюди, де використовуються електроустановки, тому приміщення без підвищеної небезпеки не можна назвати безпечними.

Електробезпека забезпечується:

- 1) відповідною конструкцією електроустановок;
- 2) застосуванням технічних способів і засобів захисту;
- 3) організаційними і технічними заходами.

Конструкція електроустановок відповідає умовам їхньої експлуатації і забезпечує захист персоналу від дотику до струмоведучих частин.

Основними технічними способами й засобами захисту від ураження електричним струмом, які використовуються окремо або в поєднанні один з одним, є:

- 1) захисне заземлення;
- 2) занулення;
- 3) вирівнювання потенціалів;
- 4) мала напруга;
- 5) електричне розділення мереж;
- 6) захисне відключення;
- 7) ізоляція струмоведучих частин;
- 8) компенсація струмів замикання на землю;
- 9) захисні пристрої;
- 10) попереджувальна сигналізація, блокування, знаки безпеки;
- 11) ізолюючі захисні і запобіжні пристосування.

Основними технічними способами і засобами захисту від ураження

електричним струмом, що передбачаються в даному дипломному проекті, є:

- 1) захисне заземлення;
- 2) занулення;
- 3) захисне відключення;
- 4) ізоляція струмоведучих частин.

Занулення в комплексі із захисним відключенням зменшує напругу дотику і обмежує час, в перебігу якого людина, торкнувшись до корпусу, може потрапити під дію напруги.

4.3 Освітлення

Трудова діяльність людини завжди протікає в певних метеорологічних умовах, які визначаються поєднанням температури повітря, швидкості його руху і відносної вологості, тиском і тепловим випромінюванням від нагрітих поверхонь. Оскільки експлуатація проектованого програмного засобу відбувається в приміщенні, то ці показники в сукупності (за винятком тиску) називаються мікрокліматом виробничого приміщення. В даний час основним нормативним документом нормалізації мікроклімату є ГОСТ 12.1.005-88 «ССБТ. Загальні санітарно-гігієнічні вимоги до повітря робочої зони» [17].

Тяжкість праці характеризує сукупну дію всіх елементів, що складають умови праці, на працездатність людини, його здоров'я, життєдіяльність і відновлення робочої сили. У такому представлені поняття тяжкості праці однаково застосовне як до розумової, так і до фізичної праці. Відповідно ГОСТ 12.1.005-88 тяжкість роботи персоналу, який обслуговує ЕОМ [22], відноситься до легкої категорії 1б (роботи, що виконуються сидячи, не вимагаючи систематичного фізичного напруження і перенесення важкостей) ГОСТ 12.1.005.88. ССБТ [17]. Загальні санітарно-гігієнічні вимоги до повітря робочої

зони. Оптимальні норми мікроклімату [20, 21] в робочій зоні, забезпечувані для робіт легкої категорії 1б приведені в табл. 4.1.

Таблиця 4.1 - Оптимальні норми мікроклімату

Період року	Температура, °С	Відносна вологість, %	Швидкість руху повітря, м/с, не більш
Холодний і перехідної	21 - 23	60 - 40	0,1
Теплий	22 - 24	60 - 40	0,2

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти) і установки кондиціонера БК-2000. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП (30 кубічних метрів на годину на одного працюючого).

Для захисту від електромагнітного випромінювання передбачаються наступні заходи:

1. застосування нових плазмових моніторів;
2. віддалення робочого місця не менше, ніж на 0,4 - 0,5 м, оскільки напруженість електричного поля зменшується при віддаленні від джерела поля;
3. встановлення раціональних режимів роботи персоналу (обмеження часу перебування);
4. раціональне розміщення в робочому приміщенні устаткування, що випромінює електромагнітну енергію.

Оскільки рівень шуму не перевищує гранично допустимих величин, які встановлені санітарними нормами, заходи для зниження шуму не проводяться [27].

Для зниження стомлюваності обслуговуючого персоналу в приміщеннях, де розташовані обчислювальні засоби, передбачається використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

У проєкті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Штучне освітлення [24] в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IV в розряді точності (середня точність). При цьому нормована освітленість на робочому місці (E_n) рівна 200 лк. Джерелом природного освітлення є сонячне світло. У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2015. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення. Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 8 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типу ЛБ (дві по 80 Вт) з

світловим потоком 5400 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (4.1):

$$N = E \cdot S \cdot Z \cdot K / F \cdot U \cdot M \quad (4.1)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300лк;

S – освітлювана площа, м²; $S = 24$ м²;

Z – поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575;

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.1), отримуємо:

$$n = (300 \cdot 30 \cdot 1.1 \cdot 1.5) / (5400 \cdot 0.575 \cdot 2) = 2.40$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.4 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Застосовують різні електричні захисні засоби від ураження струмом:

а) *ізолюючі* - ізолюють людини від струмоведучих або заземлених частин, а так-же від землі;

б) *основні* - володіють ізоляцією, здатної довго витримувати робочу напругу електроустановки і тому ними дозволяється стосуватися струмоведучих частин, що знаходяться під напругою;

в) *запобіжні* - володіють ізоляцією нездатною витримати робоча напруга електроустановки, і тому вони не можуть самостійно захищати людину від ураження струмом цим напругою. Їх значення - посилити захисні дії основних і ізолюючих засобів, разом з якими вони повинні застосовуватися, причому при використанні основних захисних засобів достатньо застосування одного заходи захисного засобу [25].

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Відповідно до класифікації приміщень за ступенем небезпеки ураження електричним струмом [18], приміщення в якому проводяться всі роботи належить до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, і 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Розрахунок проводять за допомогою методу коефіцієнта використання (екранування) електродів. Коефіцієнт використання групового заземлення η - це ставлення чинної провідності цього заземлення до найбільш можливої його провідності при нескінченно великих відстаней між його електродами. Коефіцієнт використання вертикальних заземлювачів η_v у залежності від розміщення заземлювачів і їх кількості знаходиться в межах 0,4 ... 0,99. Взаємну екрануючого дії горизонтального заземлювача (сполучної смуги)

враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c .

Послідовність розрахунку:

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = (R_d \cdot R_{пр.з.}) / (R_{пр.з.} - R_d) \quad (4.2)$$

де $R_{пр.з.}$ – опір природних заземлювачів; R_d – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_d$.

Підставивши числові значення в формулу (4.2), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом/м. Приблизне значення питомої опору глини приймаємо $\rho = 40$ Ом/м (табличне значення).

3) Розрахункова питомий опір ґрунту, $\rho_{розр.}$, Ом•м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в.}$, і горизонтальних $\rho_{розр.г.}$, Ом/м по формулі:

$$\rho_{розр.} = \psi \cdot \rho, \quad (4.3)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів і кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{розр.в.} = 1,7$ і горизонтальних $\rho_{розр.г.} = 5,5$ Ом/м.

$$\rho_{розр.в.} = 1,7 \cdot 40 = 68 \text{ Ом/м}$$

$$\rho_{розр.г.} = 5,5 \cdot 40 = 220 \text{ Ом/м}$$

4) Розраховується опір розтікання струму вертикального заземлення R_B , Ом, по (4.4).

$$R_B = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_B} \cdot \left(\ln \frac{2 \cdot l_B}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_B}{4 \cdot t - l_B} \right), \quad (4.4)$$

де l_B – довжина вертикального заземлювача (для труб - 2–3 м; $l_B=3$ м); $d_{\text{ст}}$ – діаметр стрижня (для труб - 0,03–0,05 м; $d_{\text{ст}}=0,05$ м); t – відстань від поверхні землі до середини заземлювача, яка визначається за формулою (4.5):

$$t = h_B + \frac{l_B}{2}, \quad (4.5)$$

де h_B – глибина закладення вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

$$R_B = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_B :

$$n = \frac{2 \cdot R_B}{R_d} = \frac{2 \cdot 18,5}{4} = 9,25$$

Γ визначається коефіцієнт використання вертикальних електродів групового заземлення без урахування впливу сполучної стрічки $\eta_B = 0,57$ (табличне значення).

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_B , шт:

$$n_B = \frac{2 \cdot R_B}{R_d \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16$$

7) Визначається довжина сполучної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.6)$$

де L_B – відстань між вертикальними заземлювачами, (прийняти $L_B = 3$ м);
 n_B – необхідну кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (сполучної стрічки) R_r , Ом:

$$R_r = \frac{\rho_{\text{розр.г}}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_r}, \quad (4.7)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15$ м; h_r – глибина закладення горизонтальних заземлювачів (0,5 м); l_c - довжина сполучної стрічки горизонтального заземлювача l_c , м

$$R_r = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1$$

9) Визначається коефіцієнт використання горизонтального заземлювача η_c відповідно до необхідної кількості вертикальних заземлювачів n_B .

Коефіцієнт використання сполучної смуги $\eta_c=0,3$ (табличне значення).

10) Розраховується результуючий опір заземлюючого електрода з урахуванням сполучної смуги:

$$R_{\text{заг}} = \frac{R_B \cdot R_r}{R_B \cdot \eta_c + R_r \cdot n_B \cdot \eta_B} \leq R_d. \quad (4.8)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпека будівлі [26], так як виконується умова: $R_{\text{заг}} < 4$ Ом, а саме:

$$R_{\text{заг}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_d$$

При виникненні пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як: іскри і дуги коротких замикань; перегрів провідників, резисторів і інших радіодеталей ПЕОМ, від тривалого перевантаження і наявність перехідного опору; іскри при розмиканні і розмиканні ланцюгів; розряди статичної електрики; необережне поводження з вогнем, а також вибухи газоповітряних і пароповітряних сумішей.

Важливу увагу слід звернути на пожежну безпеку підприємства в цілому і окремих його приміщень. У приміщеннях не повинно накопичуватися сміття, непотрібну папір, мотлох та ін. Речі, які не використовуються у виробничому процесі. Наявний вільний аварійний вихід за межі приміщення в разі пожежі, бути передбачені вогнегасники. Вони повинні бути в робочому стані і перевірятися відповідно до норм. У приміщеннях повинна бути пожежна сигналізація, вогнегасник. У разі виникнення пожежі необхідно повідомити в найближчу пожежну частину, убезпечити інших працівників і по можливості

прийняти кроки щодо запобігання можливих наслідків та усунення пожежі [19].

Висновок

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Було наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

ВИСНОВКИ

В даному проекті було проведено аналіз сучасних систем моніторингу транспорту, вибір обладнання, що використовується в системі для оптимальної роботи. Так, були приведені базові принципи побудови систем моніторингу, завдання моніторингу, розглянуті супутникові навігаційні системи.

Як одна з актуальших задач моніторингу є функція надання своєчасної та точної інформації про час прибуття транспортного засобу в певну контрольну точку. У зв'язку з цим реалізовано програмний комплекс для оцінки руху громадського транспорту.

Також реалізовано взаємодію програмного комплексу та фізичних об'єктів, тобто міжмашину взаємодію, що є концептуальною основою IoT. Була розроблена схема алгоритму функціонування усіх складових програмно-апаратного комплексу «InTab».

Виконано огляд існуючих аналогів для надання своєчасної інформації про час прибуття громадського транспорту, а саме завдяки інформаційному табло. Була розроблена схема збірки компонентів макету інформаційного табло.

Здійснено вибір обладнання для виготовлення макету та виклано його візуалізацію.

ПЕРЕЛІК ДжЕРЕЛ ПОСИЛАНЬ

1. Агафонов А.А. Прогнозирование параметров движения городского пассажирского транспорта по данным спутникового мониторинга / А.А. Агафонов, А.В. Сергеев, А.В. Чернов // Компьютерная оптика. – 2012. – Т. 36, №3. – С. 453-489.

2. Yao B. Path Optimization Algorithms Based on Graph Theory / B. Yao, J. Yin, H. Zhou, W. Wu // International Journal of Grid and Distributed Computing. – 2016. - Vol. 9, No. 6. - pp. 137-148.

3. Максимей И.В. Определение интегрального максимального потока в региональной сети с помощью имитационного моделирования / И.В. Максимей, Е.И. Сукач, П.Л. Гируц // Математичні машини і системи. — 2008. — № 2. — С. 128 — 136.

4. Abdullah N. Using Ford-Fulkerson Algorithm and Max Flow-Min Cut Theorem to Minimize Traffic Congestion in Kota Kinabalu, Sabah / N. Abdullah, T. K. Hua // Journal of Information System and Technology Management (JISTM). – 2017. - № 2(4). - pp. 18-34.

5. Fabbri G. An Innovative System for a Clean and Sustainable Public Transport System in Smart Cities / G.Fabbri, C.M. Medaglia, M. Ippolito, E. Saraceno, M. Antonucci, L. Fiorentino, M. Bistolfi, P. Cozzolino, M. Gallarate // In Proceedings of the 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE), Santa Clara, CA, USA. - 8–10 June 2016. - pp. 974–979.

6. Максимей И.В. Определение интегрального максимального потока в региональной сети с помощью имитационного моделирования / И.В. Максимей, Е.И. Сукач, П.Л. Гируц // Математичні машини і системи. — 2008. — № 2. — С. 128 — 136.

7. Abdullah N. Using Ford-Fulkerson Algorithm and Max Flow-Min Cut Theorem to Minimize Traffic Congestion in Kota Kinabalu, Sabah / N.

Abdullah, T. K. Hua // Journal of Information System and Technology Management (JISTM). – 2017. - № 2(4). - pp. 18-34.

8. Grinshpan A. Comparative Analysis of GPS Data / A.Grinshpan, S.Campbell // Undergraduate Journal of Mathematical Modeling: One + Two, 2014. - Vol. 5, Iss. 2.

9. Topolska K. Model of hierarchical regression in decision problems of transportation systems // Archives of Transport System. Telematics, 2017. – Vol. 10, Iss. 2.

10. Кукало И.А. Линейное отображение географических координат подвижного объекта на линейную часть магистрального нефтепровода / И.А. Кукало, С.Н. Гривцов // Известия Томского политехнического университета. Инжиниринг георесурсов, 2015. - Т. 326, № 11, С. 31–43.

11. Biswas A. Microsegmenting: An approach for precise distance calculation for GPS based ITS applications/ A.Biswas, G.Pilla, B.Tamma // IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2013.

12. Basyir M. Determination of Nearest Emergency Service Office using Haversine Formula Based on Android Platform / M.Basyir, M.Nasir, Suryati, Widdha Mellyssa // EMITTER International Journal of Engineering Technology, 2017. - Vol. 5, No. 2.

13. Berlin M. A. Safety Distance Calculation for Collision Avoidance in Vehicular Ad hoc Networks / M. A. Berlin, S. Muthusundari // Scholars Journal of Engineering and Technology (SJET), 2016. - 4(1). P. 63-69.

14. Gonçalves D. Analysis of the difference between the Euclidean distance and the actual road distance in Brazil / Daniel Neves Schmitz Gonçalves et al. // Transportation Research Procedia 3, 2014. – P. 876 – 885.

15. Manolis K., Kwistis D. Intelligent transportation systems - travelers' information systems the case of a medium size city / K. Manolis, D. Kwistis // Mechatronics, Proceedings of the IEEE International Conference on ICM '04. 2004. – pp. 200-204.

16. Chien S.I.J. Dynamic Travel Time Prediction with Real-Time and Historic Data / S.I.J. Chien C.M. Kuchipudi // Journal of Transportation Engineering. – 2003. – Vol. 129, No. 6. – pp. 608-616.

17. ГОСТ 12.1.005-88 «ССБТ. Загальні санітарно-гігієнічні вимоги до повітря робочої зони».

18. ДНАОП 0.00-1.31-99 Правила охорони праці під час експлуатації електронно-обчислювальних машин.

19. НПАОП 0.00.-1.28-10 «Правил охорони праці під час експлуатації електронно-обчислювальних машин».

20. НАПБ Б.02.005-2003 Типове положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України.

21. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» ;

22. ДСанПіН 3.3.2.007-98 «Правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

23. ГОСТ 12.1.044-89 «ССБТ. Вогнестійкість. Номенклатура показників і методи їх визначення».

24. ДБН В.2.5-28:2015 «Природне і штучне освітлення»

25. НПАОП 40.1-1.01-97 Класифікація приміщень за ступенем небезпеки ураження електричним струмом.

26. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою

27. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку.

ДОДАТОК А. Слайди презентації



**Тема дипломного проекту:
«Інформаційне табло на базі світлодіодного дисплею»**

**Студент групи КІ-14аД
Хишев В.О.
Керівник:
к.т.н, доц. Кардашук В.С.**



Актуальність



Одним з головних ознак економічної стабільності і процвітання міста є рівень розвитку транспортної системи. Єдина транспортна система повинна забезпечувати розвиток і функціонування всіх видів транспорту, і максимально відповідати потребам міста при мінімальних витратах. Що, в свою чергу, сприяє економічному, соціальному і культурному розвитку суспільства.

Завдання прогнозування часу прибуття транспортного засобу є однією з найбільш необхідною і зрозумілою для учасника дорожнього руху.



Постановка завдання



Метою даного дипломного проекту є розробка макету інформаційного табло для міської зупинки громадського транспорту.

Об'єкт дипломної роботи – програмно-технічні засоби для виготовлення інформаційного табло на базі світлодіодного дисплею.

Предмет дипломної роботи – механізми міжмашиної взаємодії, як з точки зору концептуальної основи IoT.

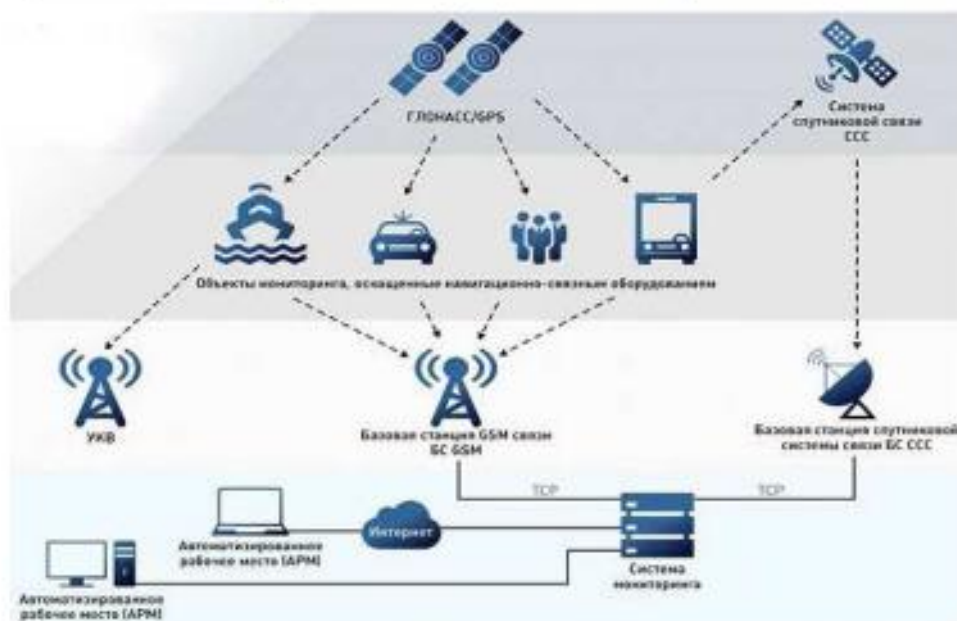
Виходячи із сучасних тенденцій на сьогоднішній день (розширення функціоналу та зменшення габаритів) виникають наступні основні **вимоги** до пристрою, що розробляється:

- забезпечення функціонування на базі контролера;
- малі габарити;
- передача отриманих даних з використанням бездротової технології.

3



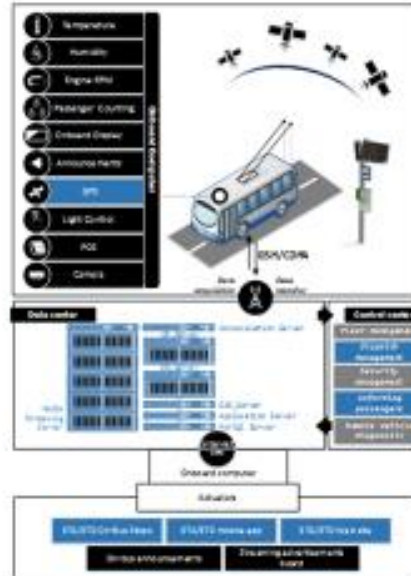
Схема моніторингу транспорту в режимі реального часу



4



Структура системи



5



Задачі розробки



- 1 **Безперервний збір навігаційної інформації про місцезнаходження транспортних засобів**
- 2 **Зберігання інформації про рух транспортних засобів в базу даних**
- 3 **Інформування пасажирів про час прибуття транспортних засобів шляхом виведення повідомлення на інформаційне табло на зупинках міста**
- 4 **Розрахунок реального часу прибуття тролейбуса на зупинку**

6



Схема передачі даних



7



Програмні засоби



Використані програмні засоби:

- PHP
- MySQL
- Apache
- ArduinoIDE

Використані протоколи:

- HTTP
- NTP
- SMTP
- URL

Використані бібліотеки:

- JSON
- cURL

Подальші плани:

- MQTT

8



Компоненти для розробки



WI-FI модуль ESP8266



Світлодіодні матриці Max7219



Usb-ttl конвертер

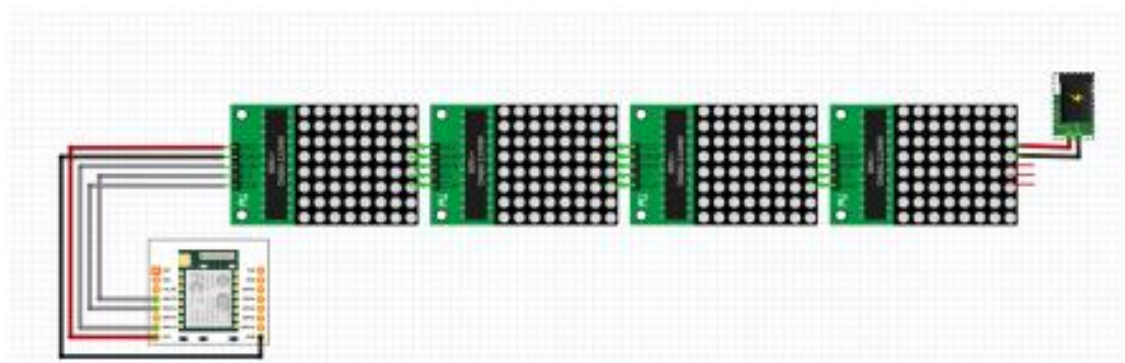


Блок живлення

9



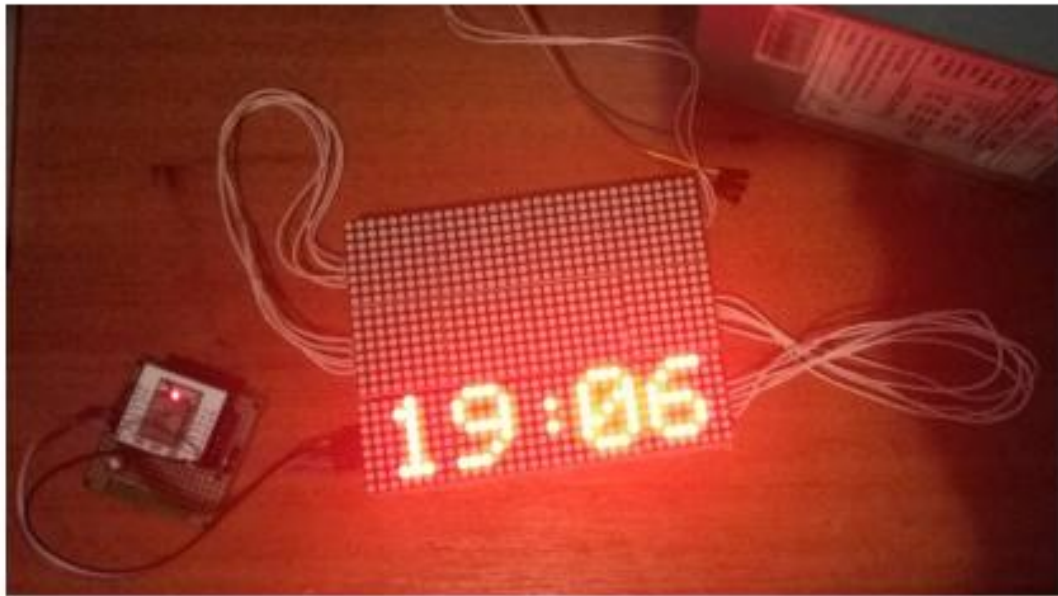
Схема підключення компонентів інформаційного табло



10



Вид інформаційного табло без корпусу



11



Наступні етапи



В подальшій
розробці буде
перехід від
MAX7219 до P10

P10

В подальшій
розробці буде
перехід від JSON
до MQTT

MQTT



12



Висновок



В результаті можна зробити висновок про те, що особливу роль в сучасних транспортних системах має функція надання своєчасної та точної інформації про час прибуття транспорту в певну точку.

Застосування апаратно-програмно комплексу є базою для функціонування інформаційних систем, які інформують диспетчерів про поточну дорожню ситуацію і пасажирів про час прибуття транспорту на зупинку, а також для автоматизації процесу формування розкладів руху пасажирського транспорту міста.

ДОДАТОК В. Код скетчу

```

1  /*
2  Подключения:
3  NodeMCU      -> Matrix
4  MOSI-D7-GPIO13  -> DIN
5  CLK-D5-GPIO14   -> Clk
6  GPIO0-D3       -> LOAD
7  */
8  #include <ESP8266HTTPClient.h>
9  #include <ESP8266WiFi.h>
10 #include <ESP8266WebServer.h>
11 #include <SPI.h>
12 #include <Adafruit_GFX.h>
13 #include <Max72xxPanel.h>
14 #include <ArduinoJson.h>
15
16 //
17 =====
18 =====
19 // Конфигурация устройства:
20 //
21 =====
22 =====
23 const char* ssid      = "your_bear_isn't_real";
24 // SSID
25 const char* password = "dthnjktN"; // пароль
26 String weatherKey = "11c9251c1ab2e5472ef6228aeeb772cf"; //
27 Чтобы получить API ключ, перейдите по ссылке
28 http://openweathermap.org/api
29 String weatherLang = "&lang=en";
30 String cityID = "691999";
31 String W_URL      = "http://api.sed.net.ua/data/1.0/time";
32 String W_API      = "11c4581cqwb2e5472gh6228kjyb294cf";
33 String W_ID       = "1";
34 //
35 =====
36 =====
37
38
39 WiFiClient client;
40
41 String weatherMain = "";
42 String weatherDescription = "";
43 String weatherLocation = "";
44 String country;
45 int humidity;
46 int pressure;
47 float temp;
48 float tempMin, tempMax;
49 int clouds;

```

```

50 float windSpeed;
51 String date;
52 String currencyRates;
53 String weatherString;
54
55 long period;
56 int offset=1,refresh=0;
57 int pinCS = 16; // Подключение пина CS
58 int numberOfHorizontalDisplays = 4;
59 int numberOfVerticalDisplays = 3;
60 String decodedMsg;
61 Max72xxPanel matrix = Max72xxPanel(pinCS,
62 numberOfHorizontalDisplays, numberOfVerticalDisplays);
63 //matrix.cp437(true);
64 String tape = "";
65 String tape_two = "";
66 int wait = 50; // скорость бегущей строки
67
68 int spacer = 2;
69 int width = 5 + spacer; // Регулируем расстояние между символами
70 const unsigned char logo [] PROGMEM = {
71     0xff, 0xff, 0xff, 0xff, 0xff, 0x0f, 0xf0, 0xff, 0xfc, 0x03,
72     0xc0, 0x3f, 0xf8, 0x01, 0x80, 0x0f,
73     0xf0, 0xe0, 0x07, 0x0f, 0xe1, 0xf8, 0x1f, 0x87, 0xe3, 0xfc,
74     0x3f, 0xc7, 0xe3, 0xfc, 0x3f, 0xc7,
75     0xe3, 0xff, 0xff, 0xc7, 0xe3, 0xff, 0xff, 0xc7, 0xf1, 0xff,
76     0xff, 0x8f, 0xf1, 0xff, 0xff, 0x8f,
77     0xf0, 0xff, 0xff, 0x0f, 0xf8, 0x7f, 0xff, 0x1f, 0xfc, 0x7f,
78     0xfe, 0x1f, 0xfc, 0x1f, 0xfc, 0x3f,
79     0xfe, 0x0f, 0xf0, 0x7f, 0xff, 0x07, 0xe0, 0xff, 0xff, 0xc1,
80     0x83, 0xff, 0xff, 0xe0, 0x07, 0xff,
81     0xff, 0xf0, 0x1f, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff, 0xff,
82     0xff, 0xff, 0xff, 0xff, 0xff, 0xff
83 };
84 const unsigned char wifi_icon [] PROGMEM = {0x07, 0xfb, 0xfd,
85 0x1e, 0xee, 0xf6, 0x36, 0xb6 };
86
87 void setup(void) {
88
89 matrix.setIntensity(13); // Яркость матрицы от 0 до 15
90
91
92 // Порядок матриц
93     matrix.setPosition(0, 0, 2); // The first display is at <0,
94 0>
95     matrix.setPosition(1, 1, 2); // The first display is at <0,
96 0>
97     matrix.setPosition(2, 2, 2); // The first display is at <0,
98 0>
99     matrix.setPosition(3, 3, 2); // The first display is at <0,
100 0>
101     matrix.setPosition(4, 3, 1); // The first display is at <0,

```



```

102 0>
103     matrix.setPosition(5, 2, 1); // The first display is at <0,
104 0>
105     matrix.setPosition(6, 1, 1); // The first display is at <0,
106 0>
107     matrix.setPosition(7, 0, 1); // The first display is at <0,
108 0>
109     matrix.setPosition(8, 0, 0); // The first display is at <0,
110 0>
111     matrix.setPosition(9, 1, 0); // The first display is at <0,
112 0>
113     matrix.setPosition(10, 2, 0); // The first display is at <0,
114 0>
115     matrix.setPosition(11, 3, 0); // The first display is at <0,
116 0>
117 // Ориентация каждой матрицы
118     matrix.setRotation(0, 1); // The first display is position
119 upside down
120     matrix.setRotation(1, 1); // The first display is position
121 upside down
122     matrix.setRotation(2, 1); // The first display is position
123 upside down
124     matrix.setRotation(3, 1); // The first display is position
125 upside down
126     matrix.setRotation(4, 3); // The first display is position
127 upside down
128     matrix.setRotation(5, 3); // The first display is position
129 upside down
130     matrix.setRotation(6, 3); // The first display is position
131 upside down
132     matrix.setRotation(7, 3); // The first display is position
133 upside down
134     matrix.setRotation(8, 1); // The first display is position
135 up
136 openweathermap.org
137 Weather API
138 Simple and fast and free weather API from OpenWeatherMap you
139 have access to current weather data, 5- and 16-day forecasts, UV
140 Index, air pollution
141 and historical data
142 side down
143     matrix.setRotation(9, 1); // The first display is position
144 upside down
145     matrix.setRotation(10, 1); // The first display is
146 position upside down
147     matrix.setRotation(11, 1); // The first display is
148 position upside down
149     matrix.fillScreen(LOW);
150     matrix.drawBitmap(0, 0, logo, 32, 24, 0, 1);
151     matrix.write();
152     delay(5000);
153

```

```

154     Serial.begin(9600);                                     // Дебаг
155     WiFi.mode(WIFI_STA);
156     WiFi.begin(ssid, password);                             //
157     Подключаемся к WIFI
158     while (WiFi.status() != WL_CONNECTED) {                // Ждем до
159     посинения
160         matrix.drawBitmap(12, 8,  wifi_icon, 8, 8, 0, 1);
161         matrix.write();
162         delay(500);
163         Serial.print(".");
164     }
165
166
167
168 }
169
170 //
171 =====
172 =====
173 #define MAX_DIGITS 16
174 byte dig[MAX_DIGITS]={0};
175 byte digold[MAX_DIGITS]={0};
176 byte digtrans[MAX_DIGITS]={0};
177 int updCnt = 0;
178 int dots = 0;
179 long dotTime = 0;
180 long clkTime = 0;
181 int dx=0;
182 int dy=0;
183 byte del=0;
184 int h,m,s;
185 //
186 =====
187 =====
188 void loop(void) {
189
190     matrix.fillRect(0, 0, 32, 8, LOW);
191
192
193     if(updCnt<=0) { // каждые 10 циклов получаем данные времени и
194     погоды
195         updCnt = 10;
196         Serial.println("Getting data ...");
197         getWeatherData();
198         getTime();
199         GetTrolleyTime();
200         Serial.println("Data loaded");
201         clkTime = millis();
202     }
203
204     if(millis()-clkTime > 15000 && !del && dots) { //каждые 15
205     секунд запускаем бегущую строку

```

```

206 ScrollText(utf8rus(weatherString));
207     updCnt--;
208     clkTime = millis();
209 }
210
211
212
213
214 DisplayTime();
215     if(millis()-dotTime > 500) {
216         dotTime = millis();
217         dots = !dots;
218     }
219
220
221 }
222
223 //
224 =====
225 =====
226 void DisplayTime(){
227     updateTime();
228     matrix.fillScreen(LOW);
229     int y = 16; // Центрируем текст по Вертикали
230
231
232     if(s & 1){matrix.drawChar(14, y, (String(":"))[0], HIGH,
233 LOW, 1);} //каждую четную секунду печатаем двоеточие по центру
234 (чтобы мигало)
235     else{matrix.drawChar(14, y, (String(" "))[0], HIGH, LOW,
236 1);}
237
238     String hour1 = String (h/10);
239     String hour2 = String (h%10);
240     String min1 = String (m/10);
241     String min2 = String (m%10);
242
243     int xh = 2;
244     int xm = 19;
245 //     int xs = 28;
246
247     matrix.drawChar(xh, y, hour1[0], HIGH, LOW, 1);
248     matrix.drawChar(xh+6, y, hour2[0], HIGH, LOW, 1);
249     matrix.drawChar(xm, y, min1[0], HIGH, LOW, 1);
250     matrix.drawChar(xm+6, y, min2[0], HIGH, LOW, 1);
251 matrix.drawChar(0, 0, tape[0], HIGH, LOW, 1);
252 matrix.drawChar(0, 0, tape[1], HIGH, LOW, 1);
253 matrix.drawChar(0, 0, tape[2], HIGH, LOW, 1);
254 matrix.drawChar(0, 0, tape[3], HIGH, LOW, 1);
255 matrix.drawChar(0, 0, tape[4], HIGH, LOW, 1);
256 matrix.drawChar(0, 0, tape[5], HIGH, LOW, 1);
257 matrix.drawChar(0, 0, tape[6], HIGH, LOW, 1);

```

```

258     matrix.drawChar(0, 0, tape[7], HIGH, LOW, 1);
259
260     matrix.drawChar(0, 8, tape_two[0], HIGH, LOW, 1);
261     matrix.drawChar(0, 8, tape_two[1], HIGH, LOW, 1);
262     matrix.drawChar(0, 8, tape_two[2], HIGH, LOW, 1);
263     matrix.drawChar(0, 8, tape_two[3], HIGH, LOW, 1);
264     matrix.drawChar(0, 8, tape_two[4], HIGH, LOW, 1);
265     matrix.drawChar(0, 8, tape_two[5], HIGH, LOW, 1);
266     matrix.drawChar(0, 8, tape_two[6], HIGH, LOW, 1);
267     matrix.drawChar(0, 8, tape_two[7], HIGH, LOW, 1);
268
269
270
271     matrix.write(); // Вывод на дисплей
272 }
273
274 //
275 =====
276 =====
277 void DisplayText(String text){
278     matrix.fillScreen(LOW);
279     for (int i=0; i<text.length(); i++){
280
281         int letter =(matrix.width())- i * (width-1);
282         int x = (matrix.width() +1) -letter;
283         int y = 16; // Центрируем текст по Вертикали
284         matrix.drawChar(x, y, text[i], HIGH, LOW, 1);
285         matrix.drawChar(0, 0, tape[0], HIGH, LOW, 1);
286     matrix.drawChar(0, 0, tape[1], HIGH, LOW, 1);
287     matrix.drawChar(0, 0, tape[2], HIGH, LOW, 1);
288     matrix.drawChar(0, 0, tape[3], HIGH, LOW, 1);
289     matrix.drawChar(0, 0, tape[4], HIGH, LOW, 1);
290     matrix.drawChar(0, 0, tape[5], HIGH, LOW, 1);
291     matrix.drawChar(0, 0, tape[6], HIGH, LOW, 1);
292     matrix.drawChar(0, 0, tape[7], HIG
293 H, LOW, 1);
294
295     matrix.drawChar(0, 8, tape_two[0], HIGH, LOW, 1);
296     matrix.drawChar(0, 8, tape_two[1], HIGH, LOW, 1);
297     matrix.drawChar(0, 8, tape_two[2], HIGH, LOW, 1);
298     matrix.drawChar(0, 8, tape_two[3], HIGH, LOW, 1);
299     matrix.drawChar(0, 8, tape_two[4], HIGH, LOW, 1);
300     matrix.drawChar(0, 8, tape_two[5], HIGH, LOW, 1);
301     matrix.drawChar(0, 8, tape_two[6], HIGH, LOW, 1);
302     matrix.drawChar(0, 8, tape_two[7], HIGH, LOW, 1);
303
304     matrix.write(); // Вывод на дисплей
305
306     }
307
308 }
309

```

```

310 //
311 =====
312 =====
313 void ScrollText (String text){
314     for ( int i = 0 ; i < width * text.length() + matrix.width()
315 - 1 - spacer; i++ ) {
316         if (refresh==1) i=0;
317         refresh=0;
318         matrix.fillScreen(LOW);
319         int letter = i / width;
320         int x = (matrix.width() - 1) - i % width;
321         int y = 16; // Центрируем текст по Вертикали
322
323         while ( x + width - spacer >= 0 && letter >= 0 ) {
324             if ( letter < text.length() ) {
325                 matrix.drawChar(x, y, text[letter], HIGH, LOW, 1);
326                 matrix.drawChar(0, 0, tape[0], HIGH, LOW, 1);
327                 matrix.drawChar(0, 0, tape[1], HIGH, LOW, 1);
328                 matrix.drawChar(0, 0, tape[2], HIGH, LOW, 1);
329                 matrix.drawChar(0, 0, tape[3], HIGH, LOW, 1);
330                 matrix.drawChar(0, 0, tape[4], HIGH, LOW, 1);
331                 matrix.drawChar(0, 0, tape[5], HIGH, LOW, 1);
332                 matrix.drawChar(0, 0, tape[6], HIGH, LOW, 1);
333                 matrix.drawChar(0, 0, tape[7], HIGH, LOW, 1);
334
335                 matrix.drawChar(0, 8, tape_two[0], HIGH, LOW, 1);
336                 matrix.drawChar(0, 8, tape_two[1], HIGH, LOW, 1);
337                 matrix.drawChar(0, 8, tape_two[2], HIGH, LOW, 1);
338                 matrix.drawChar(0, 8, tape_two[3], HIGH, LOW, 1);
339                 matrix.drawChar(0, 8, tape_two[4], HIGH, LOW, 1);
340                 matrix.drawChar(0, 8, tape_two[5], HIGH, LOW, 1);
341                 matrix.drawChar(0, 8, tape_two[6], HIGH, LOW, 1);
342                 matrix.drawChar(0, 8, tape_two[7], HIGH, LOW, 1);
343
344             }
345             letter--;
346             x -= width;
347         }
348         matrix.write(); // Вывод на дисплей
349         delay(wait);
350     }
351 }
352
353
354 //
355 =====
356 =====
357 // Берем погоду с сайта openweathermap.org
358 //
359 =====
360 =====
361

```

```

362
363
364 const char *weatherHost = "api.openweathermap.org";
365
366 void getWeatherData()
367 {
368     Serial.print("connecting to "); Serial.println(weatherHost);
369     if (client.connect(weatherHost, 80)) {
370         client.println(String("GET /data/2.5/weather?id=") + cityID
371 + "&units=metric&appid=" + weatherKey + weatherLang + "\r\n" +
372         "Host: " + weatherHost + "\r\nUser-Agent:
373 ArduinoWiFi/1.1\r\n" +
374         "Connection: close\r\n\r\n");
375     } else {
376         Serial.println("connection failed");
377         return;
378     }
379     String line;
380     int repeatCounter = 0;
381     while (!client.available() && repeatCounter < 10) {
382         delay(500);
383         Serial.println("w.");
384         repeatCounter++;
385     }
386     while (client.connected() && client.available()) {
387         char c = client.read();
388         if (c == '[' || c == ']') c = ' ';
389         line += c;
390     }
391
392     client.stop();
393
394     DynamicJsonBuffer jsonBuf;
395     JsonObject &root = jsonBuf.parseObject(line);
396     if (!root.success())
397     {
398         Serial.println("parseObject() failed");
399         return;
400     }
401     //weatherMain = root["weather"]["main"].as<String>();
402     weatherDescription =
403     root["weather"]["description"].as<String>();
404     weatherDescription.toLowerCase();
405     // weatherLocation = root["name"].as<String>();
406     // country = root["sys"]["country"].as<String>();
407     temp = root["main"]["temp"];
408     humidity = root["main"]["humidity"];
409     pressure = root["main"]["pressure"];
410     tempMin = root["main"]["temp_min"];
411     tempMax = root["main"]["temp_max"];
412     windSpeed = root["wind"]["speed"];
413     clouds = root["clouds"]["all"];

```

```

414     String deg = String(char('~'+25));
415     weatherString = "t: " + String(temp,1)+" ";
416     weatherString += weatherDescription;
417     weatherString += " Влажн.: " + String(humidity) + "% ";
418     weatherString += "Давл.: " + String(pressure/1.3332239) + " мм
419 ";
420 // weatherString += "Облачность: " + String(clouds) + "% ";
421     weatherString += "Ветер: " + String(windSpeed,1) + "М/с";
422 }
423
424 // =====
425 =====
426 // Берем время у GOOGLE
427 //
428 =====
429 =====
430
431 float utcOffset = 3; //поправка часового пояса
432 long localEporc = 0;
433 long localMillisAtUpdate = 0;
434
435 void getTime()
436 {
437     WiFiClient client;
438     if (!client.connect("www.google.com", 80)) {
439         Serial.println("connection to google failed");
440         return;
441     }
442
443     client.print(String("GET / HTTP/1.1\r\n") +
444                 String("Host: www.google.com\r\n") +
445                 String("Connection: close\r\n\r\n"));
446     int repeatCounter = 0;
447     while (!client.available() && repeatCounter < 10) {
448         delay(500);
449         //Serial.println(".");
450         repeatCounter++;
451     }
452
453     String line;
454     client.setNoDelay(false);
455     while(client.connected() && client.available()) {
456         line = client.readStringUntil('\n');
457         line.toUpperCase();
458         if (line.startsWith("DATE: ")) {
459             date = "      "+line.substring(6, 22);
460             h = line.substring(23, 25).toInt();
461             m = line.substring(26, 28).toInt();
462             s = line.substring(29, 31).toInt();
463             localMillisAtUpdate = millis();
464             localEporc = (h * 60 * 60 + m * 60 + s);
465         }

```

```

466     }
467     client.stop();
468 }
469 //
470 =====
471 =====
472 void GetTrolleyTime() {
473     HTTPClient client;
474     String url = W_URL;
475     url += "?id=";
476     url += W_ID;
477     url += "&APPID=";
478     url += W_API;
479     url += "&units=metric&lang=ru";
480     Serial.println(url);
481     client.begin(url);
482     int httpCode = client.GET();
483     if( httpCode == HTTP_CODE_OK ) {
484         String httpString = client.getString();
485         ParseTrolleyTime(httpString);
486     }
487     client.end();
488 //
489 =====
490 =====
491 }
492 void ParseTrolleyTime(String s) {
493     DynamicJsonBuffer jsonBuffer;
494     JsonObject& root = jsonBuffer.parseObject(s);
495
496     if (!root.success()) {
497         Serial.println("Json parsing failed!");
498         return;
499     }
500     tape = "6 ";
501     tape_two = "2 ";
502     double sec = root["sixto"]["sec"];
503     if( sec > 60) tape += "> 1 min";
504     else if( sec < 60 && sec >=120 ) tape += "> 2 min ";
505     else if( sec < 120 && sec >=180 ) tape += "> 3 min ";
506     else if( sec < 180 && sec >=240 ) tape += "> 4 min ";
507     else if( sec < 240 && sec >=300 ) tape += "> 5 min ";
508     else if( sec < 300 && sec >=360 ) tape += "> 6 min ";
509     else if( sec < 360 && sec >=420 ) tape += "> 7 min ";
510     else if( sec < 420 && sec >=480 ) tape += "> 8 min ";
511     else if( sec < 480 && sec >=540 ) tape += "> 9 min ";
512     else if( sec < 540 && sec >=600 ) tape += "> 10 min ";
513     else if( sec < 600 && sec >=660 ) tape += "> 11 min ";
514     else if( sec < 660 && sec >=720 ) tape += "> 12 min ";
515     else if( sec < 720 && sec >=780 ) tape += "> 13 min ";
516     else if( sec < 780 && sec >=840 ) tape += "> 14 min ";
517     else if( sec < 840 && sec >=900 ) tape += "> 15 min ";

```



```

518     double sec1 = root["twoto"]["sec"];
519     if( sec > 60)tape += "> 1 min";
520     else if( sec1 < 60 && sec1 >=120 )tape_two += "> 2 min ";
521     else if( sec1 < 120 && sec1 >=180 )tape_two += "> 3 min ";
522     else if( sec1 < 180 && sec1 >=240 )tape_two += "> 4 min ";
523     else if( sec1 < 240 && sec1 >=300 )tape_two += "> 5 min ";
524     else if( sec1 < 300 && sec1 >=360 )tape_two += "> 6 min ";
525     else if( sec1 < 360 && sec1 >=420 )tape_two += "> 7 min ";
526     else if( sec1 < 420 && sec1 >=480 )tape_two += "> 8 min ";
527     else if( sec1 < 480 && sec1 >=540 )tape_two += "> 9 min ";
528     else if( sec1 < 540 && sec1 >=600 )tape_two += "> 10 min ";
529     else if( sec1 < 600 && sec1 >=660 )tape_two += "> 11 min ";
530     else if( sec1 < 660 && sec1 >=720 )tape_two += "> 12 min ";
531     else if( sec1 < 720 && sec1 >=780 )tape_two += "> 13 min ";
532     else if( sec1 < 780 && sec1 >=840 )tape_two += "> 14 min ";
533     else if( sec1 < 840 && sec1 >=900 )tape_two += "> 15 min ";
534     s = "";
535 }
536
537 //
538 =====
539 =====
540
541 void updateTime()
542 {
543     long curEpoch = localEpoc + ((millis() - localMillisAtUpdate)
544 / 1000);
545     long epoch = round(curEpoch + 3600 * utcOffset + 86400L) %
546 86400L;
547     h = ((epoch % 86400L) / 3600) % 24;
548     m = (epoch % 3600) / 60;
549     s = epoc
550 h % 60;
551 }
552
553 //
554 =====
555 =====
556
557
558 String utf8rus(String source)
559 {
560     int i,k;
561     String target;
562     unsigned char n;
563     char m[2] = { '0', '\0' };
564
565     k = source.length(); i = 0;
566
567     while (i < k) {
568         n = source[i]; i++;
569

```

```
570     if (n >= 0xC0) {
571         switch (n) {
572             case 0xD0: {
573                 n = source[i]; i++;
574                 if (n == 0x81) { n = 0xA8; break; }
575                 if (n >= 0x90 && n <= 0xBF) n = n + 0x30-1;
576                 break;
577             }
578             case 0xD1: {
579                 n = source[i]; i++;
580                 if (n == 0x91) { n = 0xB8; break; }
581                 if (n >= 0x80 && n <= 0x8F) n = n + 0x70-1;
582                 break;
583             }
584         }
585     }
586     m[0] = n; target = target + String(m);
587 }
588 return target;
589 }
```