

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)  
(рівень вищої освіти)

Система управління безпекою на залізничних переїздах  
(тема)

Виконав:  
здобувач 4 року навчання,  
групи КІУКІ-21-7

Олексенко І. Р.  
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма

Комп'ютерна інженерія  
(повна назва освітньої програми)

Керівник ас. Безродний В.В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

Чумаченко С.В.  
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія  
(шифр і назва)

Тип програми Освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Олексенко Івану Романовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Система управління безпекою на залізничних переїздах

затверджена наказом по університету від " 21 " 05 2025 р. № 403 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 10.06.2025

3. Вихідні дані до роботи (проекту) \_\_\_\_\_

Мікроконтролери STM32F030C8T6, Intel MCS 51

САПР Poteus

Мова програмування C

Середовище розробки Keil for ARM

4. Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

Обладнання та системи безпеки на залізничних переїздах

Технологічна платформа реалізації

Реалізація електричної схеми мкіроконтролерного пристрою

Середовище розробки програмного забезпечення

Програмування мкіроконтролерного пристрою

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) \_\_\_\_\_  
16 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

7. Дата видачі завдання 7.05.2025

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Видача теми проекту, узгодження і затвердження	01.05.2025 -03.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	03.05.2025 -10.05.2025	
3	Розробка моделі пристрою керування	10.05.2025 -15.05.2025	
4	Розробка електричної схеми пристрою	15.05.2025 -20.05.2025	
5	Розробка програми для мікроконтролера	20.05.2025 -30.05.2025	
6	Проведення макетування пристрою	31.05.2025 -07.06.2025	
7	Оформлення пояснювальної записки	08.06.2025 -12.06.2025	
8	Перевірка виконаного проекту керівником	12.06.2025 -15.06.2025	
9	Захист проекту	15.06.2025 -20.06.2025	

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ ас. Безродний В.В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка містить 49 сторінок, 24 рисунки, 9 джерел за переліком посилань.

ЗАЛІЗНИЧНИЙ ПЕРЕЇЗД, СИСТЕМА АВТОМАТИКИ, КІНЦЕВИЙ АВТОМАТ, ГРАФ ПЕРЕХОДІВ, МІКРОКОНТРОЛЕР, STM32F030C8T6, МОВА ПРОГРАМУВАННЯ C, IDE KEIL FOR ARM.

Розроблений прототип пристрою керування автоматичним залізничним переїздом з використанням механічних датчиків наближення потягу. Для реалізації пристрою керування застосований керуючий автомат Мура, математична модель якого представлена у формі темпорального графа переходів з використанням гіперстану та обробника зовнішніх подій.

Керуючий автомат реалізований на базі мікроконтролера STM32F030C8T6. Програмна реалізація алгоритму керування здійснена на мові програмування C у середовищі розробки Keil for ARM. Моделювання розробленого пристрою керування проводилося інструментальними засобами САПР Proteus.

## ABSTRACT

Bachelor's thesis contains 49 pages, 24 figures, 9 sources according to the list of links.

RAILWAY CROSSING, AUTOMATIC SYSTEM, FINITE STATE MACHINE, STATE DIAGRAM, MICROCONTROLLER, STM32F030C8T6, C PROGRAMMING LANGUAGE, IDE KEIL FOR ARM.

Developed a prototype of an automatic railroad crossing control device using mechanical sensors for approaching a train

The prototype of the automatic railroad crossing control device using mechanical sensors for approaching a train is developed. To implement the control device, Moore's control finite state machine, the mathematical model of which is represented in the form of a temporal state diagram using the hyperstate and the handler of external events is used.

The controller is implemented on the basis of the microcontroller STM32F030C8T6. The software implementation of the control algorithm is implemented in the programming language C in the Keil for ARM development environment. The simulation of the developed control device was carried out by the instrumental CAD tools of Proteus.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
1 ПРИСТРОЇ КЕРУВАННЯ БЕЗПЕКОЮ НА ЗАЛІЗНИЧНИХ ПЕРЕЇЗДАХ .....	10
1.1 Обладнання та системи безпеки на залізничних переїздах .....	10
1.2 Розрахунок параметрів системи безпеки.....	13
1.3 Автоматні пристрої керування .....	15
1.4 Технічне завдання на проектування.....	18
2 ТЕХНОЛОГІЧНА ПЛАТФОРМА РЕАЛІЗАЦІЇ.....	20
2.1 Модель пристрою керування залізничним переїздом.....	20
2.2 Мікроконтролер STM32F030C8T6.....	23
2.3 Електрична схема мікроконтролерного пристрою.....	26
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРИСТРОЮ КЕРУВАННЯ.....	28
3.1 Середовище розробки програмного забезпечення.....	28
3.2 Програмна модель пристрою керування та її моделювання.....	32
3.3 Макетування пристрою керування залізничним переїздом.....	40
ВИСНОВКИ.....	47
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	48
ДОДАТОК А Графічна частина проекту.....	50
ДОДАТОК Б Код програми пристрою керування автоматичним залізничним переїздом на базі МК MCS51.....	58
ДОДАТОК В Код програми пристрою керування автоматичним залізничним переїздом на базі МК STM32F030C8T6.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ

АПШ – автоматичний переїзний шлагбаум;

ДТП – дорожньо-транспортна пригода;

КА – керуючий автомат;

МК – мікроконтролер;

САПР – система автоматизованого проектування;

РЕП – радіоелектронні пристрої;

ТПВ – таблиця переходів-виходів;

ЦП – цифровий пристрій;

ARM – Advanced RISC Machine, (вдосконалена RISC-машина);

ASIC – application-specific integrated circuit, (інтегральна схема спеціального призначення);

GUI – graphical user interface, (графічний інтерфейс користувача);

IDE – Integrated Development Environment (інтегроване середовище розробки);

IoT – Internet of Things, (Інтернет речей);

RISC – Reduced Instruction Set Computer (комп'ютер (процесор) зі скороченим набором команд);

SoC – System of Chip (Система на Кристалі ).

## ВСТУП

Перетин автомобільних доріг із залізничними коліями в одному рівні є найбільш складними і небезпечними елементами дорожньої мережі, які надають значний вплив на ефективність експлуатації автомобільного та залізничного транспорту в цілому.

Проблема залізничних переїздів є дуже важливою ще й тому, що більше половини від загального числа переїздів становлять регульовані переїзди, розташовані на маршрутах основних пасажирських перевезень. Дорожньо-транспортні пригоди (ДТП) на цих переїздах супроводжуються, як правило, великою кількістю постраждалих.

Залізничні переїзди, як об'єкти підвищеної небезпеки, на яких залізничний транспорт має перевагу в русі перед усіма іншими видами транспорту, згідно з вимогами нормативних актів залізничного транспорту [1] повинні бути обладнані необхідними пристроями, що забезпечують безпеку руху, поліпшують умови пропуску поїздів і транспортних засобів.

Однією з першочергових завдань у вирішенні проблеми зниження ризику виникнення ДТП на залізничних переїздах є вдосконалення технічної оснащеності ділянок автомобільних доріг в межах переїздів і на підходах до них. Попередження аварійності на залізничних переїздах має здійснюватися на основі підготовки нової та перегляду чинної нормативно-технічної документації, шляхом розробки та впровадження на переїздах нових технічних засобів забезпечення безпеки з використанням нової елементної бази [2]. В залежності від інтенсивності руху на переїздах застосовують огорожувальні пристрої у вигляді автоматичної світлофорної сигналізації; автоматичної переїзної сигналізації з автоматичними шлагбаумами; автоматичної сповіщувальної сигналізації.

Незважаючи на стрімкий розвиток технологій, темпи впровадження мікроелектронних і мікропроцесорних систем і компонентів критичного

застосування замість існуючих релейних в даний час залишаються невисокими. Основними причинами, які стримують впровадження цих систем і компонентів, є їх висока вартість, необхідність підготовки фахівців, які могли б успішно обслуговувати дані системи, а також недоліки, властиві застосовуваним методам і засобам їх побудови. Зокрема, значні складності у розробників виникають при розробці засобів сполучення мікроелектронних структур з виконавчими механізмами. Розробка систем керування автоматичними залізничними переїздами на сучасній елементній базі є актуальною задачею.

Протягом останніх років одним з головних напрямків в області розробки мікроконтролерів був розвиток мікроконтролерів ARM. На сьогоднішній день є близько 240 ARM-мікроконтролерів від численних виробників. Компанія ST Microelectronics (європейська мікроелектронна компанія, одна з найбільших, які займаються розробкою, виготовленням і продажем різних напівпровідникових електронних і мікроелектронних компонентів) випустила перші мікроконтролери на базі нового ядра ARM Cortex-M3, STM32. Ці пристрої встановлюють нові стандарти продуктивності і ціни, здатності виконання алгоритмів управління жорсткого реального часу при низькому енергоспоживанні.

Таким чином, актуальною є задача розробки мікроконтролерної системи управління рухом на залізничному переїзді на базі мікроконтролера STM32.

# 1 ПРИСТРОЇ КЕРУВАННЯ БЕЗПЕКОЮ НА ЗАЛІЗНИЧНИХ ПЕРЕЇЗДАХ

## 1.1 Обладнання та системи безпеки на залізничних переїздах

Залізничні переїзди, тобто перетин залізничних колій та автомобільних доріг на одному рівні є найбільш складними і небезпечними елементами транспортної мережі, які суттєво впливають на безпеку експлуатації автомобільного й залізничного транспорту в цілому. Передусім ці перетини характеризуються непродуктивними затримками автотранспорту. Однак найгострішою проблемою залишаються дорожньо-транспортні пригоди на переїздах – у тому числі з особливо тяжкими наслідками і значними людськими жертвами.

Переїзди поділяються на регульовані та ті, що не регулюються. До регульованих належать переїзди, які обладнані пристроями переїзної сигналізації або такі, що обслуговуються черговим працівником та обладнані шлагбаумами. Регульовані переїзди, що обладнані пристроями переїзної сигналізації з автоматичними шлагбаумами, обслуговуються черговим працівником або на них встановлюють систему відеонагляду.

Переїзди повинні розташовуватися на прямих ділянках колії та автомобільних доріг за межами виїмок та місць, де не забезпечуються достатні умови видимості. Перехрещення залізничних колій з автомобільними дорогами повинно виконуватися переважно під прямим кутом. При неможливості виконання цієї умови гострий кут між коліями та автомобільною дорогою має бути не менше 60 град. Забороняється влаштування переїздів через станційні колії, де здійснюється організоване приймання та відправлення або безперервний рух поїздів. Такі переїзди необхідно закривати або переносити на перегін.

Автоматична світлофорна сигналізація – система, що складається з

переїзних світлофорів звукової сигналізації, електричних рейкових кіл і апаратури керування і яка забезпечує автоматичне включення сповіщення про наближення поїзда до переїзду та включення світлофорної і звукової сигналізації. Після проходження поїзда за переїзд і звільнення переїзду хвостом поїзда сигналізація вимикається автоматично. На окремих переїздах з автобусним рухом переїзна сигналізація доповнюється місячно-білими мигаючими вогнями на переїзних світлофорах. .

Ділянка наближення – розташована перед переїздом ділянка колії, довжина якої визначається розрахунком у залежності від швидкості руху поїздів та довжини проїзної частини переїзду, і обладнана рейковими колами для завчасного автоматичного сповіщення на переїзд про наближення до нього поїзда і автоматичного керування переїзною сигналізацією і шлагбаумами, якщо переїзд ними обладнаний;

Шлагбаум – пристрій, який перекриває проїзну частину автомобільної дороги для припинення руху транспортних засобів через залізничний переїзд. Шлагбауми бувають автоматичні, напівавтоматичні та механічні.

Шлагбаум - це пристрій, який дозволяє перекрити проїзд шириною 2-12 м за рахунок стріли, що повертається у вертикальній або горизонтальній площині. Як правило, вони використовуються там, де не висуваються високі вимоги щодо захисту від проникнення, але необхідний контроль в'їзду автомобілів на територію. Автоматичні шлагбауми застосовуються на в'їздах на автостоянки, платні дороги, автопарки і на територію ін.

На відміну від будь-яких типів воріт, що забезпечують повне загородження зони проїзду, шлагбаум лише позначає контрольну лінію, проїзд якої без сигналів, що дозволяють, або відповідного пропуску заборонено. Сучасні обмежувачі проїзду не бояться вітрів, дощів та морозів, оскільки всі деталі конструкції виконані із стійких до погодних умов матеріалів. Тому вони зберігають свою працездатність у будь-яку пору року.

В даний час у більшості випадків на переїздах використовуються автоматичні шлагбауми Конструктивно вони складаються з двох основних

елементів - стійки, в яку вбудований двигун і блок керування, і стріли. Додатково шлагбауми можуть бути оснащені підставкою для стріли, фотоелементами, спідницею та іншими елементами безпеки та керування.

Стійка (тумба) є основою шлагбауму, на яку кріпляться стріла та силовий механізм (електродвигун, пружина, що балансує, та ін.). Корпус стійки, як правило, виготовляється із стійкого до механічних пошкоджень та корозії антивандального матеріалу.

Стріла шлагбауму призначена для безпосереднього перекриття отвору. Часто стріли виготовляються з алюмінію, проте бувають і з нержавіючої сталі, дерева та спеціальних сплавів. За формою вони бувають прямокутного, трикутного та круглого перерізу. Довжина стріли визначається шириною проїзду, який необхідно перекрити.

Підставка для стріли або уловлювача є необов'язковим елементом для стріл невеликої довжини. Однак вона незамінна для зупинки та утримування стріли при закритті шлагбауму та для стійкості конструкції у закритому стані під час вітру та інших зовнішніх впливів при довгій стрілі або обтяженій додатковими елементами, такими як знак дорожнього руху, спідниця тощо.

Спідниця – ще один додатковий елемент конструкції, призначений для запобігання перетину проїзду під стрілою шлагбауму та для забезпечення більшої помітності обмежувача проїзду водіями транспортних засобів в умовах утрудненої видимості (опади, туман, темний час доби).

Рекомендується оснащувати автоматичний шлагбаум системою захисту від наїзду на перешкоду – комплектом фотоелементів, що також дозволить захистити транспортний засіб від опускання шлагбауму під час здійснення проїзду. В цілому, автоматичні шлагбауми прості в керуванні та незамінні в рамках систем контролю та управління доступом або як самостійні елементи обмеження проїзду для наведення порядку та позбавлення суєти, а також від попадання на територію об'єкта небажаних транспортних засобів

Для автоматичних шлагбаумів переведення загороджувальних брусів

шлагбаумів у закрите (горизонтальне) положення здійснюється автоматично через розрахунковий час після вступу поїзда на ділянку наближення і включення червоних вогнів переїзних світлофорів. Переведення у відкрите (вертикальне) положення здійснюється також автоматично після проходження поїзда за переїзд і звільнення переїзду хвостом поїзда. При цьому червоні вогні переїзних світлофорів гаснуть.

Для напівавтоматичних шлагбаумів переведення загороджувальних брусів шлагбаумів у закрите (горизонтальне) положення здійснюється автоматично через розрахунковий час після вступу поїзда на ділянку наближення чи відкриття світлофора. Для механізованих шлагбаумів існує механічний привід, за допомогою якого черговий по переїзду вручну переводить загороджувальні бруси у відкрите (вертикальне) або закрите (горизонтальне) положення.

На відстані не менше ніж 2,5 м від крайньої рейки на узбіччі автодороги розташовуються сигнальні огорожування (стовпчики) висотою 0,75-0,8 м. На підходах до переїздів з боку залізничної колії повинні бути встановлені постійні попереджувальні сигнальні знаки "С" про подавання машиністами поїздів свистка, а з боку автомобільної дороги на переїздах без чергового - дорожні знаки 1.29 "Одноколійна залізниця" або 1.30 "Багатоколійна залізниця". Сигнальні знаки "С" встановлюються з правого боку відповідно до напрямку руху поїздів на відстані 500-1500 м, а на переїздах, де рухаються поїзди із швидкістю понад 120 км/год, - на відстані 800-1500 м від переїздів.

## 1.2 Розрахунок параметрів системи безпеки

Відповідно до вимог Інструкції з улаштування та експлуатації залізничних переїздів [2] автоматична сигналізація переїзду повинна забезпечувати подачу сигналу зупинки у бік автомобільної дороги, а автоматичні шлагбауми - приймати закрите положення за час, необхідний

для завчасного звільнення переїзду транспортними засобами до підходу поїзду до переїзду. Необхідно, щоб автоматична світлофорна сигналізація продовжувала діяти до повного звільнення поїздом переїзду.

Переїзд має закриватися своєчасно, для цього необхідний розрахунок норм часу до підходу поїзду та швидкості поїзду на ділянці колії, де знаходиться переїзд. Визначимо необхідний час сповіщення про наближення поїзду до переїзду:  $T_p = T_1 + T_2 + T_3$ , де  $T_1$  – час, необхідний автомобілю для того, що проїхати переїзд.  $T_2$  – час спрацювання апаратури, з;  $T_3$  – гарантійний запас часу.

Визначимо довжину ділянки наближення:  $L_p = 0.28 * V_{max} * T_p$ , де 0,28 – коефіцієнт переведення швидкості з км/ч в м/с,  $V_{max}$  – максимальна швидкість руху поїздів, яка дозволена на цій ділянці. По встановлених нормах час сповіщення про наближення поїзду до переїзду має бути не менше 40 секунд, при автоматичній світлофорній сигналізації на переїзді з автоматичними шлагбаумами (АПС) та не менше 5 с. після перетинання переїзду останнім вагоном поїзду. На рис. 1.1 наведена схема розрахунку довжини ділянки наближення до залізничного переїзду.

При швидкості  $V=72$  км/год (20 м/сек),  $T_p = 40$ с.  $T_в = 5$ с, тобто  $L_p = 0.28 * 72 * 40 = 800$  м,  $L_в = 0.28 * 72 * 5 = 100$  м,

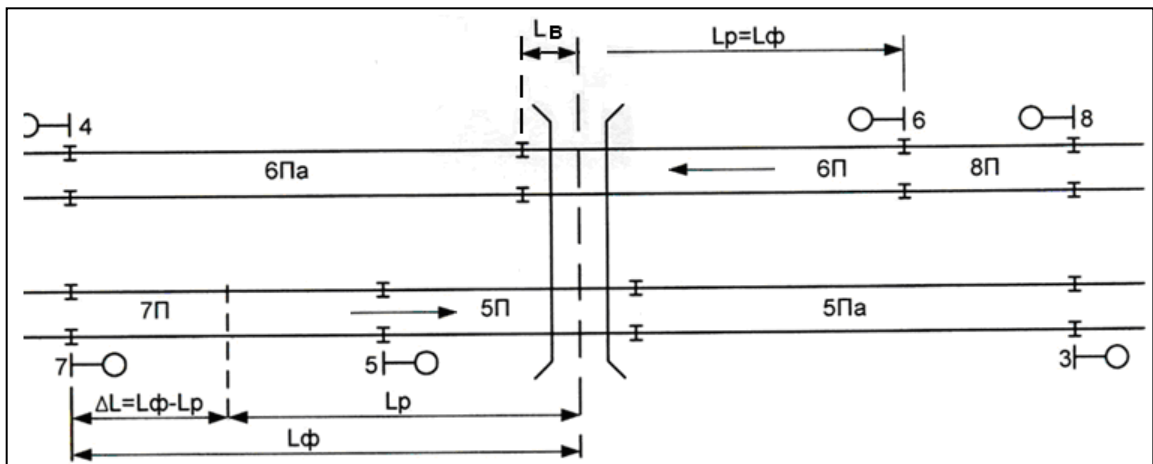


Рисунок 1.1 – Схема розрахунку довжини ділянки наближення до залізничного переїзду

Для фіксації наближення поїзду до залізничного переїзду використовують датчики різного типу, а саме, ємнісного типу, оптичного типу, механічного типу тощо.

Найбільш поширеними є контактні датчики з парою електричних контактів, які замикаються рухливим транспортним засобом, що чинить тиск на цей датчик. Роль такого датчика найчастіше виконує стандартна кнопка-вимикач. Нині контактні датчики застосовуються практично в усіх областях техніки: машинобудування, авіація, охоронні системи, автоматичні електронні пристрої тощо. При макетуванні пристрою керування буде використовуватися кнопка, яка імітує роботу контактного датчика.

Найбільш простими і надійними є контактні датчики реакційного типу, виконані у вигляді двох ізольованих контактів. Один з контактів виконаний у вигляді центрального електроду, жорстко сполученого з зовнішнім корпусом, а інший - у вигляді зовнішньої оболонки, розміщеної відносно центрального електроду з проміжком. При цьому зовнішня оболонка є чутливим елементом датчика. При вчиненні тиску на зовнішню оболонку, відбувається замикання контактів датчика (рис.1.2).



Рисунок. 1.2 – Універсальний вологозахисений контактний датчик, регульований по висоті, довжина 63 мм StarLine PS-8

### 1.3 Автоматні пристрої керування

Серед множини керуючих пристроїв можна виділити пристрої

логічного управління, у яких керуючі сигнали (control value) представляються в двійковому алфавіті. Оскільки для реалізації керуючої частини в таких пристроях, як правило, використовуються кінцеві автомати, ці автомати називаються керуючі автомати (КА).

При описі алгоритму функціонування цифрових пристроїв логічного управління САПР цифрових пристроїв одним із стилів написання коду є стиль автоматного програмування. Автоматні програми строго структуровані і в них виділено три види функцій: функції переходів, функції виходів, функції реалізації затримок та переходу до нового стану. Автоматні програми строго шаблонізовані з використанням операторів багатопозиційного вибору (switch, case), умовних операторів (if, select) та функцій реалізації таймера або фронту (синхросигналу Clk). Автоматні програми інваріантні способу кодування (мов програмної реалізації) [3] .

Попередній аналіз проблеми побудови пристроїв керування показує, що як основа його моделі може бути використана модель керуючого автомата. Проте пряме використання КА для моделювання складних пристроїв керування неможливо, оскільки, з одного боку, її привабливі властивості (можливість відбити стійкість, подієвість і циклічність) явно не підкреслені, а з іншого боку, простий КА не припускає операцій з часовими інтервалами і паралелізм у системах реального часу. Для усунення цього протиріччя запропонована формальна модель реалізації складного алгоритму управління у вигляді гіперавтомата.

Таким чином можна визначити, що гіперавтомат – це автомат, який має у своєму складі гіперстани. Гіперстан – сукупність станів автомату, переходи між які здійснюються безумовно з урахуванням тільки часу існування кожного із станів. Стани автомата, які входять до гіперстану, повністю або частково зациклені. Вхід до гіперстану відбувається завжди до першого стану з множини станів гіперстану. Вихід з гіперстану відбувається або по закінченні часу існування гіперстану, або при наявності «зовнішньої» події (які формується поза множиною вихідних оповіщувальних сигналів

автомату), незалежно від того, в якому зі станів гіперстану знаходиться автомат у цей момент часу. У гіперстані можуть змінюватися оповіщувальні сигнали основного автомату, але вихід з гіперстану здійснюється або за параметром часу, або за зовнішнім сигналом (подією), який не є оповіщувальним для будь-якого зі станів автомату. Зображення гіперстану на графі переходів автомата показано на рис. 1.3.

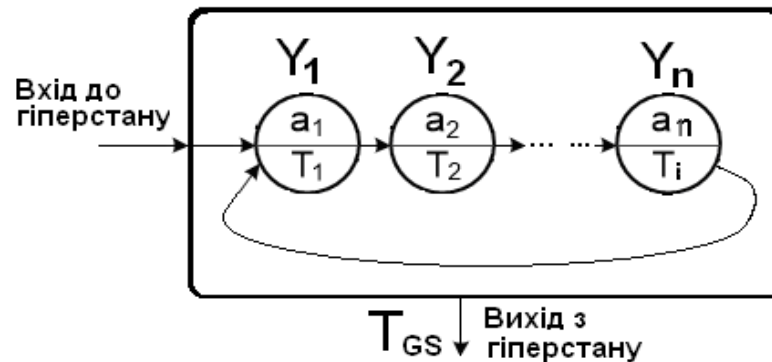


Рисунок 1.3 – Графічна модель гіперстану у темпоральному графі переходів автомату

У системах логічного управління поняття «значення вхідних сигналів» (input values), поділяється на вхідні дії (input actions), та події (events). Вхідні дії реалізуються автоматично шляхом опитування відповідно до алгоритму його роботи в циклі управління, а події реалізуються миттєво та призводять до зміни стану автомата.

Подія – це абстрактне поняття, що передбачає таку зміну умов довкілля, що породжує певну реакцію системи. Події можуть генеруватися як довкіллям, і усередині самої системи управління її компонентами.[4].

Існують три основні варіанти взаємодії керуючого автомата із зовнішнім середовищем.

1. Події використовуються для взаємодії керуючого та операційного автоматів усередині автоматної системи управління. У цьому випадку, якщо події є винятковими (дві події не можуть відбуватися одночасно), обробка

подій нічим не відрізняється від обробки значень вхідних змінних автомата.

2. Події поряд із вхідними змінними забезпечують взаємодію автомата із зовнішнім середовищем. Це проектне рішення має відображати відмінність між подіями та вхідними змінними: автомат обробляє події в момент його появи, тоді як значення вхідних змінних опитуються автоматом за його власною ініціативою.

3. Кожній події зіставляється окремий стан (перехід) в автоматі. Це рішення підходить лише для реалізації виключної моделі подій. Крім того, воно відображає активну роль подій і той факт, що саме виникнення подій ініціює роботу автомата.

#### 1.4 Технічне завдання на проектування

Метою проекту є розробка макету мікроконтролерного пристрою керування безпекою автоматичного залізничного переїзду.

Система безпеки автоматичного залізничного переїзду складається з двох електромеханічних датчиків (датчика наближення поїзда до переїзду та датчика проїзду переїзду поїздом), двох червоних ліхтарів, звукового оповіщувача та електромеханічного шлагбауму. Існуючі приклади реалізації таких систем безпеки залізничних переїздів наведені на рис. 1.4.

Алгоритм роботи пристрою керування автоматичним залізничним переїздом наступний. При відсутності поїзда в зоні безпеки залізничного переїзду ліхтарі не горять, звукового сигналу немає, шлагбаум знаходиться у піднятому стані. При наближенні поїзда спрацьовує датчик наближення, включається миготіння червоних ліхтарів, включається звуковий оповіщувач і через визначений проміжок часу опускається шлагбаум. Звукові та світлові сигнали подаються до того часу, поки поїзд не залишить зону безпеки залізничного переїзду. Спрацьовує датчик проїзду переїзду поїздом, ліхтарі та сповіщувач виключаються, а шлагбаум підіймається. Система безпеки переходить до режиму очікування наближення наступного поїзду.



Рисунок 1.4 – Технічна реалізація системи безпеки автоматичного залізничного переїзду

Пристрій керування має використовувати дешевий та надійний тип мікроконтролера. Макет пристрою має бути зібраний на платі налаштування, а всі виконавчі дії мають моделюватися з використанням кольорових світлодіодів.

## 2 ТЕХНОЛОГІЧНА ПЛАТФОРМА РЕАЛІЦІИ

### 2.1 Модель пристрою керування залізничним переїздом

Як абстрактну модель дискретного пристрою логічного управління будемо використовувати кінцевий автомат, який визначається шістькою ,  $F = \langle X, A, Y, \delta, \lambda, a_1 \rangle$  де  $X = \{x_1, x_2, \dots, x_m\}$  – множина букв вхідного алфавіту;  $A = \{a_1, a_2, \dots, a_n\}$  – множина станів автомата;  $Y = \{y_1, y_2, \dots, y_r\}$  – множина букв вихідного алфавіту,  $\delta(a_i, x_k) = a_j$  – функція переходів автомата,  $\lambda(z_i, x_k) = y_\alpha$  – функція виходів автомата,  $a_1$  – початковий стан автомата. Однією з форм представлення моделі абстрактного автомата є граф переходів (state diagram).

При переході від моделі абстрактного автомата до схемної реалізації пристрою керування використовується модель структурного автомата. У структурних моделях вхідний алфавіт перетворюється на множинув вхідних сигналів (set input values), алфавіт станів перетворюється на множинув внутрішніх змінних (internal variables), а вихідний алфавіт – на множинув вихідних сигналів (set output values). При цьому всі три множини є кінцевими. Загальноприйнята модель структурного автомата (модель Хаффмена, рис. 2.1), складається з комбінаційного та послідовного компонентів.

Граф переходів керуючого автомата є візуальним поданням моделі не тільки абстрактного, але й структурного автомата. Такий граф переходів, за аналогією зі змістовною граф-схемою алгоритму (ГСА), називатимемо змістовним графом переходів кінцевого автомата. Таким чином, змістовний темпоральний граф переходів є не лише візуальним відображенням алгоритму роботи структурного автомата, а й його повною математичною моделлю. Змістовий граф переходів у САПР описується шаблонами

спеціального виду (patterns) мовами програмування чи мовами опису апаратури [4].

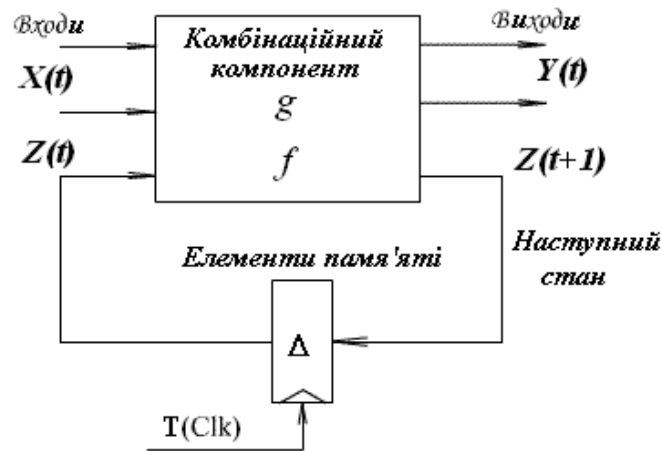


Рисунок 2.1 – Модель структурного автомату

Виходячи з моделей абстрактного та структурного автоматів формалізуємо алгоритм управління автоматичним залізничним переїздом [5].

При наближенні поїзду до залізничного спрацьовує датчика D1, сигнал про наближення поїзду по дротяному, або іншому виду зв'язку, подається до пристрою (рис. 2.2 а). Пристрій керування, умовно кажучи, приймає рішення про закриття переїзду для автомобільного транспорту. Команда про закриття переїзду подається на засіб світлової сигналізації (світлофор), який змінює дозволяючий на забороняючий. Шлагбаум закривається. Після того, як останній вагон поїзду перетнув район залізничного переїзду, спрацьовує датчик D2, який подає відповідний сигнал до пристрою керування, який з урахуванням часових інтервалів здійснює процедуру відкриття залізничного переїзду для автомобільного транспорту.

Множина вхідних сигналів:  $X = \{ On, D1, D2 \}$ , де  $On = \{0, 1\}$  – сигнал включення пристрою керування,  $D1 = \{0, 1\}$  – сигнали датчика наближення поїзда до переїзду,  $D2 = \{0, 1\}$  – сигнали датчика проїзду переїзду поїздом.

Множина вихідних сигналів:  $Y = \{ L1, L2, P1, P2, Rg \}$ , де  $L1$  – червоне світло лівого світлофора,  $L2$  – червоне світло правого світлофора,

$P1$  – сигнал опускання шлагбаума,  $P2$  – сигнал підняття шлагбаума,  $Rg$  – сигнал звукового сповіщувача.

Графічне зображення інтерфейсу пристрою керування автоматичним залізничним переїздом наведено на рис. 2.2 (а).

Сформулюємо алгоритм роботи пристрою керування автоматичним залізничним переїздом в терміна станів керуючого автомату. Після включення системи безпеки ( $Op = 1$ ) КА переходить зі стану  $a1$  у стан  $a2$ . При відсутності поїзду в зоні безпеки залізничного переїзду КА знаходиться у стані  $a2$ , при цьому ліхтарі не горять, звукового сигналу немає, шлагбаум знаходиться у піднятому стані ( $L1=0, L2=0, P1=1, P2=0$ ). При наближенні поїзду спрацьовує датчик наближення ( $D1=1$ ) і автомат переходить до стану  $a3$ . При цьому включаються обидва червоні ліхтарі та опускається шлагбаум ( $L1=1, L2=1, P1=0, P2=1$ ). Після реалізації затримки на закриття шлагбаума ( $T2$ ) автомат переходить до гіперстану включається миготіння червоних ліхтарів та звуковий оповіщувач Гіперстан реалізується шляхом зациклювання станів  $a4$  та  $a5$  на час  $T4$ , або до приходу сигналу  $D2=1$ . Звукові та світлові сигнали подаються до того часу, поки поїзд не залишить зону безпеки залізничного переїзду. Спрацьовує датчик проїзду переїзду поїздом ( $D2=1$ ), автомат переходить до стану  $a6$ , ліхтарі та оповіщувач виключаються, а шлагбаум підіймається ( $L1=0, L2=0, P1=1, P2=0$ ). Після відпрацювання затримки  $T2$  система безпеки переходить до режиму чекання наступного поїзду.

Для реалізації керуючого автомату оберемо часовий автомат Мура, бо вихідний сигнал пристрою керування залежить тільки від станів автомату, а його тривалість визначається внутрішньою затримкою певного стану часового автомату.

Граф переходів часового гіперавтомату, керуючого автоматичним залізничним переїздом, зображений на рис. 2.2 (б).

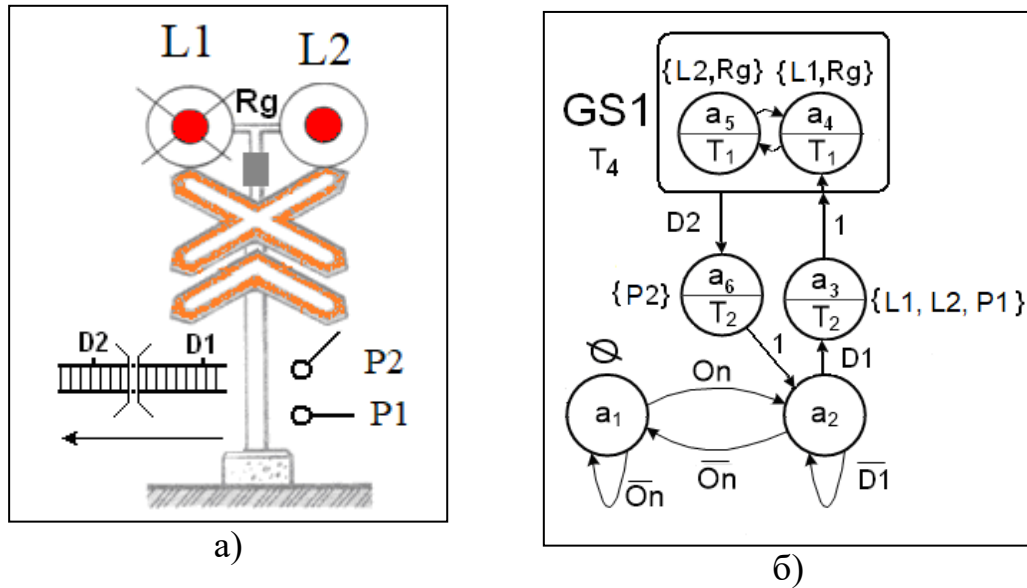


Рисунок 2.2 – Інтерфейс та граф переходів пристрою керування автоматичним залізничним переїздом

Пристрій, який розробляється є системою реального часу, який функціонує на основі подій, тому треба при програмуванні мікроконтролерної системи керування детально описати часові параметри роботи пристрою з урахуванням алгоритму обробки події.

Відмітимо, що затримки  $T1$  – миготіння червоного світла світлофора (1 сек.),  $T2$  – час на закриття (відкриття) шлагбаума (10 сек), які реалізуються на рівні програмування мікроконтролерного (МК) пристрою, а миготіння червоно світла реалізується за рахунок введення гіперстану GS1 з затримкою  $T4$ , яка достатньо велика та може перелаштовуватися..

## 2.2 Мікроконтролер STM32F030C8T6

Вибір мікроконтролера обумовлений тим, що STM32f3 реалізований на ядрі ARM Cortex має малий трафаретний розмір чіпа, мале енергоспоживання в перерахунку на одну команду в MIPS/мВт, наявність великої кількості периферії, що дозволяє вирішувати дуже велику різноманітність задач без підключення додаткових модулів. Також STM32f3

має високу продуктивність (швидкість виконання команд).

У системах реального часу з жорстким обмеженням на час відгуку, де висуваються високі вимоги щодо продуктивності, необхідні 32-бітові мікроконтролери, такі як STM32f3. Досить низька вартість, зручність програмування та наявність безкоштовного програмного забезпечення сприяли його просуванню

Архітектура мікроконтролерів ARM (від англ. Advanced RISC Machine - вдосконалена RISC-машина) – це сімейство ліцензованих 32-бітних і 64-бітових мікропроцесорних ядер розробки компанії ARM Limited. ARM – британська корпорація, один з найбільших розробників і ліцензіарів архітектури 32-розрядних і 64-розрядних RISC-процесорів, орієнтованих на використання в портативних і мобільних пристроях [6].

Мікроконтролери серії STM32F030C8T6 компанії STMicro інтегрують 32-бітний ARM-процесор Cortex™ -M0 з робочою частотою до 48 МГц, високошвидкісну пам'ять (до 64 Кбайт FLASH і до 8 Кбайт SRAM), функціональний блок периферійних модулів і ліній введення-виводу. Всі пристрої серії підтримують стандартний набір комунікаційних інтерфейсів (до двох каналів I2C, до двох каналів SPI і до двох UART), один 12-бітний АЦП, до шести 16-бітових таймерів загального призначення і ШІМ-таймер з розширеними можливостями управління. Мікроконтролери розраховані на роботу в діапазоні температур навколишнього середовища від -40 °С до +85 °С при напрузі живлення від 2.4 В до 3.6 В. Кілька режимів дозволяють створювати додатки з малим енергоспоживанням.

Галузі застосування STM32F030: портативне обладнання, аудіо-відео ресивери та цифрові телевізори, периферійний пристрої ПК, ігрові платформи, двоколісні транспортні засоби з електроприводом, побутова техніка, принтери, сканери, системи сигналізації, пристрої відеоконференцзв'язку, системи клімат-контролю.

У табл. 2.1 наведені основні параметри МК STM32F030C8T6, а на рис. 2.3 – контакти корпусу цієї мікросхеми .

Таблиця 2.1 – Основні параметри МК STM32F030C8T6

Ядро	Cortex-M0
Ширина шини даних	32 біт
Максимальна тактова частота	48 МГц
Корпус	LQFP 48
Розмір пам'яті FLASH	64 кб
Розмір ОЗП даних	8 кб
Кількість таймерів	5
Інтерфейси	2xSPI ; 2xI2C ; 2xUSART
Робоча напруга	2,4 - 3,6 В.
Робоча температура	-40...+85

Ядро Cortex-M0, на сьогоднішній день, є самим «маленьким» зі всієї великої кількості ядер ARM. Ядро характеризується виключно малою площею на кристалі, низьким споживанням і дуже високою щільністю коду, що дозволяє досягати 32-бітної продуктивності при вартості 8/16-бітних рішень.

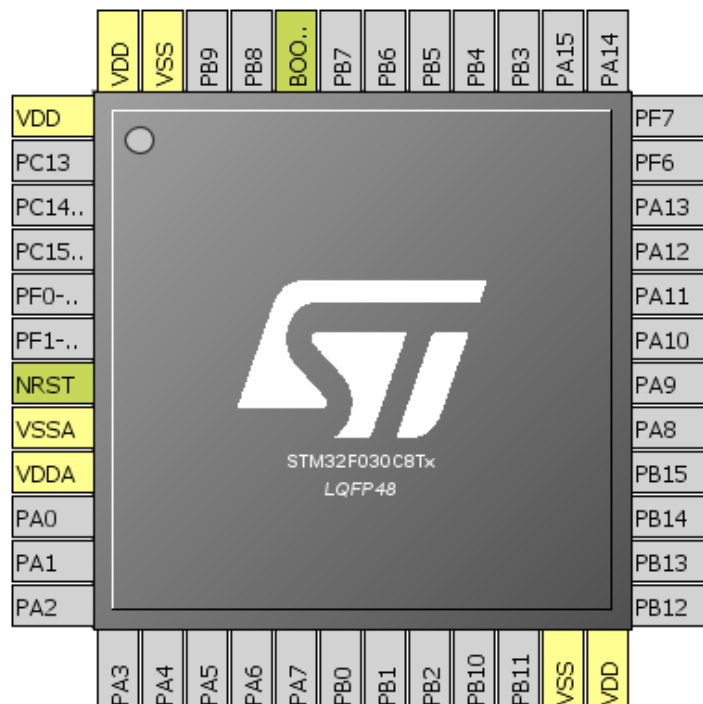


Рисунок 2.3 – Контакти корпусу мікросхеми STM32F030C8Tx

### 2.3 Електрична схема мікроконтролерного пристрою

Для розробки та моделювання електричної принципової схеми пристрою керування залізничним переїздом будемо використовувати систему автоматизованого проектування радіоелектронних пристроїв (САПР РЕП) Proteus [7].

Процес проектування цифрових пристроїв (ЦП) прийнято розбивати на етапи (системний, схемний, конструкторський, технологічний), а сам спроектований РЕП на рівні (система, підсистема або апаратура, прилад, блок, осередок або вузол). Виходячи з такої декомпозиції, представляється природною вимога, щоб САПР підтримували усі етапи і рівні проектування в повному об'ємі.

Proteus – це комерційний пакет програм класу САПР, об'єднуючий у собі дві основні програми: ISIS – засіб розробки і налаштування у режимі реального часу електронних схем і ARES – засіб розробки друкованих плат. Як автоматичний трасувальник може використовуватися допоміжна програма "ELECTRA Autorouter". Програма "Proteus" має великі бібліотеки компонентів, у тому числі і периферійних пристроїв: індикації, температурних датчиків, інтерактивних елементів введення-виводу : кнопок, перемикачів, віртуальних портів і віртуальних вимірювальних приладів, інтерактивних графіків, які не завжди присутні в інших подібних програмах. Імітація реальної поведінки електронних компонентів і схем, складених з них, в програмі ISIS ґрунтується на застосуванні мови SPICE – моделювання поведінки компонентів. Мова SPICE (Simulation Program with Integrated Circuit Emphasis) була розроблена в Каліфорнійському університеті Берклі на початку сімдесятих років минулого століття і де-факто став основою для більшості програм, що моделюють поведінку електронних компонентів. Це і P-CAD і MultiSim (Workbench) і навіть OrCAD. Відмінність Proteus від згаданих САПР полягає в тому, що якщо там використовується версія

PSPICE (розроблена фірмою MicroSim Corporation, яку в 1998 році благополучно "проковтнув" OrCAD), то ISIS ProSPICE базується на іншому клоні, іменованому SPICE3F5.

Особливістю САПР Proteus є те, що в її бібліотеках знаходяться в основному мікросхеми виробництва відомих фірм USA. Мікросхеми STM32F030C8T6 в бібліотеках наявної версії Proteus немає, тому для моделювання розробленої схеми будемо використовувати одну з мікросхем сімейства Intel MCS 51, а саме AT89C52.

Створена в САПР Proteus електрична схема пристрою керування залізничним переїздом показана на рис. 2.4.

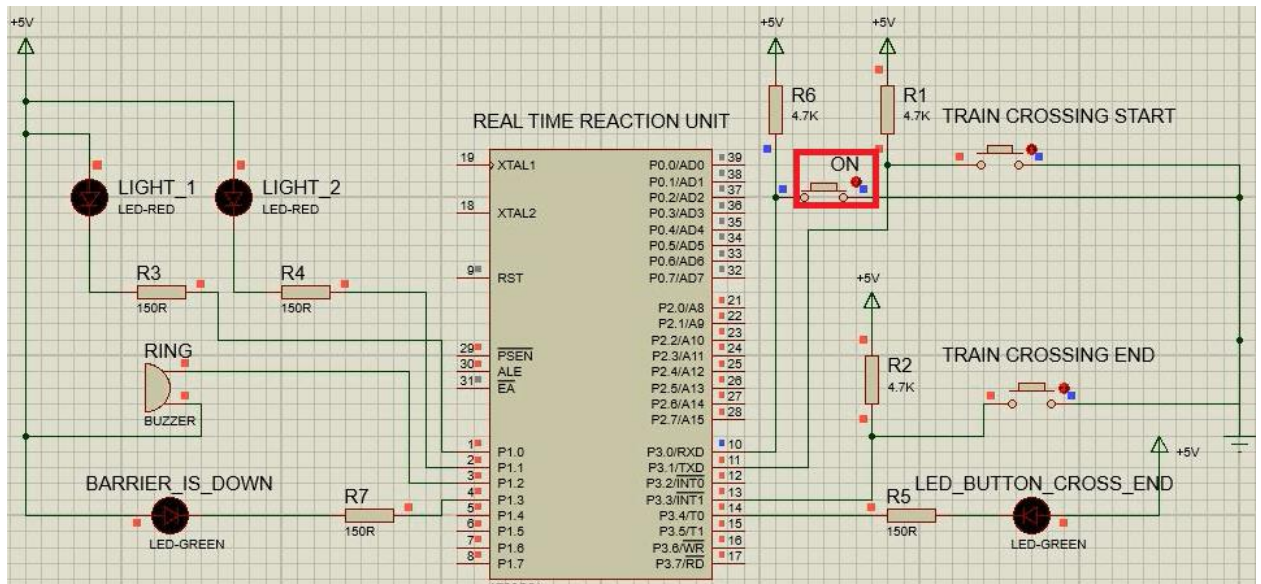


Рисунок 2.4 – Структурна схема пристрою керування залізничним переїздом на базі МК MCS 51

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРИСТРОЮ КЕРУВАННЯ

### 3.1 Середовище розробки програмного забезпечення

У сучасному світі все більшого поширення набувають системи на основі вбудованих процесорів призначених для ефективного виконання вузького класу задач в умовах жорстких обмежень на співвідношення продуктивності, енергоспоживання, розміру та вартості виготовлення.

У процесі створення вбудованих систем найважливішу роль відіграє інструментарій розробки, що дозволяє виконувати розробку, налагодження і профілювання програм для цільової системи (пристроїв на основі вбудованих процесорів) з використанням інструментальної машини. Основними компонентами такого інструментарію є асемблер, компоувальник, симулятор, налагоджувач, профілювальник. В якості інструментальної машини, як правило, виступає звичайна робоча станція.

Метою використання крос-інструментів є створення на інструментальній машині файлу з двійковим образом початкового вмісту пам'яті (як машинні команди, так і дані) для пристроїв на основі вбудованих процесорів. Такий файл потім використовується для завантаження в конкретні цільові пристрої.

Звичайно процес взаємодії програміста з крос-інструментами відбувається через візуальне інтегроване середовище розробки (IDE) з вбудованим редактором, механізмами підтримки проєктів, різними засобами керування й аналізу результатів роботи окремих інструментів. Реальні програми звичайно складаються з декількох модулів, кожному з яких відповідає файл із вихідними текстами програм (мовами високого рівня або асемблера).

Звичайно процес взаємодії програміста з крос-інструментами відбувається через візуальне інтегроване середовище розробки (IDE) з

вбудованим редактором, механізмами підтримки проектів, різними засобами керування й аналізу результатів роботи окремих інструментів. Реальні програми звичайно складаються з декількох модулів, кожному з яких відповідає файл із вихідними текстами програм (мовами високого рівня або асемблера).

Компілятор транслює модулі мовою високого рівня в проміжні асемблерні модулі. Асемблер відповідає за перетворення асемблерних модулів (як написаних вручну, так і згенерованих компілятором) в об'єктні модулі з машинними кодами і даними для цільової апаратури.

Компонувальник виконує складання декількох об'єктних модулів в один абсолютний модуль з об'єднанням відповідних секцій вхідних об'єктних файлів. При цьому виконуються переміщення символів за абсолютними адресами пам'яті (автоматично або відповідно до заданого програміста картою пам'яті) з відповідними виправленнями залежних від них кодів команд і значень даних. На цьому закінчується етап складання програми.

Отриманий абсолютний модуль можна перетворити в образ пам'яті для безпосереднього завантаження в цільову апаратуру за допомогою програматора. Для програміста взаємодія з моделлю апаратури здійснюється через налагоджувач, який дозволяє переглядати стан моделі (уміст пам'яті, регістрів, шин, сигналів) і здійснювати кероване (у тому числі, покрокове) виконання цільової програми на рівні окремих команд або рядків вихідного коду. Разом з налагоджувачем використовуються різні види профілювальників і засобів аналізу для візуалізації отриманих характеристик моделі.

Як модель цільової системи для крос-налагоджувача найчастіше виступає симулятор, що дозволяє моделювати цільову апаратуру повністю на інструментальній машині. Існують стимулятори різного рівня абстракції від функціонального симулятора на рівні всієї системи до симуляторів на рівні системи команд і симуляторів, емулюючих точну структуру апаратури на рівні функціональних блоків і конкретних вентилів. У складі інструментарію

крос-розробки звичайно використовується симулятор рівня системи команд (у тому числі, з урахуванням конвеєрних ефектів з потактовою точністю)

Середовище розробки Keil є набором утиліт для виконання повного комплексу заходів щодо написання програмного забезпечення для мікроконтролерів. Базовим для Keil є поняття моделі контролера. Модель контролера визначає сукупність апаратних ресурсів, які враховуються компілятором та симулятором у складі системи. Кожна модель контролера відповідає апаратній структурі певної модифікації фізичного мікроконтролера сімейства MCS-51. Таким чином, модель мікроконтролера - це сукупність наступних параметрів:

- обсяги внутрішньої та зовнішньої пам'яті програм та даних;
- номенклатура та кількість компонентів периферії, що використовуються в даній модифікації мікроконтролера;
- параметри конфігурації, що використовуються при симуляції, наприклад, розмір і швидкість доступу для моделі пам'яті EEPROM.

Програма працює на ПК під керуванням тільки ОС Windows версій Win\_10 та вище.

Особливою відмінністю Keil for ARM є бібліотека мікроконтролерів сімейства STM, яка доступна розробнику. Ця бібліотека містить всі мікроконтролери сімейства STMicroelectronics, при встановленні якої, розробник має доступ до програмування цих пристроїв.

Компанія STM надає бібліотеку драйверів периферійних пристроїв, бібліотеку для USB (як ANSI C і бібліотеку) і вихідні коди, сумісні з попередніми бібліотеками для мікроконтролерів STM7 і STM9. Уже доступні адаптовані версії цих бібліотек для популярних компіляторів. Крім цього, для сімейства Cortex є багато відкритих і комерційних RTOS і сполучного ПЗ (TCP / IP, файлові системи і т.д.). У Cortex-M3 використовується абсолютно нова система налагодження під назвою CoreSight. Доступ до системи CoreSight здійснюється через порт Dedug Access, який підтримує з'єднання або через стандартний JTAG, або через послідовний інтерфейс serial wire.

Загальний вигляд вікна середовища Keil for ARM та його бібліотеки представлені на рис. 3.1, 3.2.

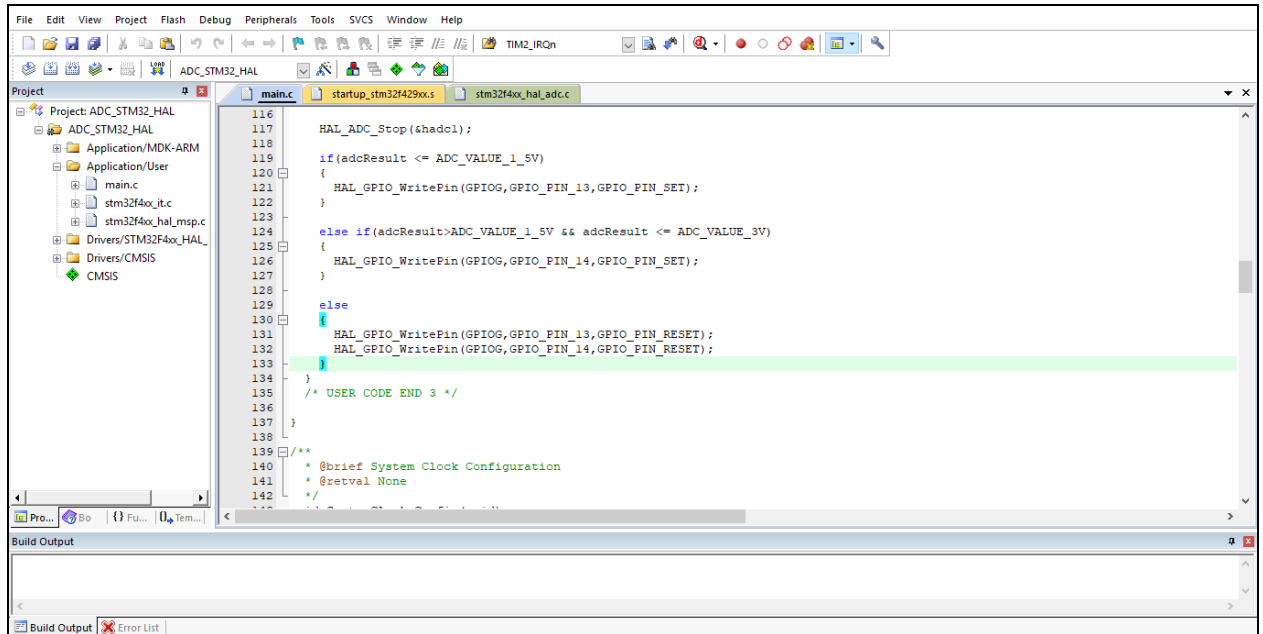


Рисунок 3.1 – Загальне вигляд вікна середовища розробки Keil for ARM

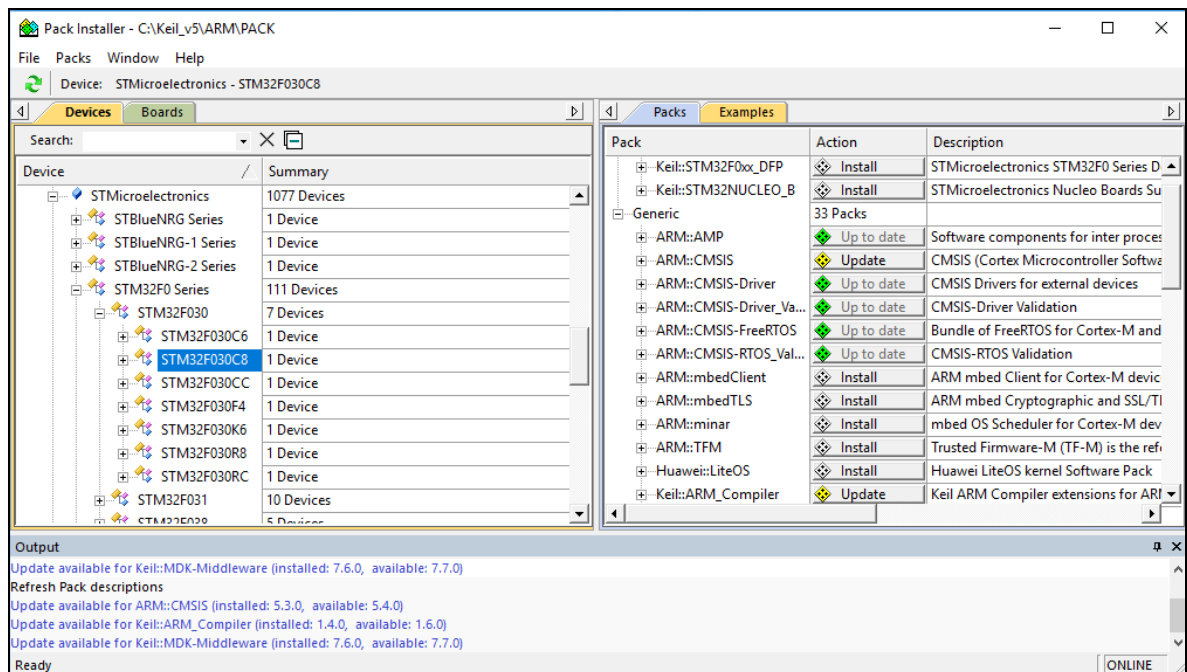


Рисунок 3.2 – Бібліотека мікроконтролерів STMicroelectronics та компоненти, які необхідно встановити для розробки пристрою

### 3.2 Програмна модель пристрою керування та її моделювання

Програма реалізації пристрою керування залізничним переїздом на базі мікроконтролера STM32F030C8T6 виконана на мові програмування C у середовищі розробки Keil for ARM.

Код програми на мові C для МК STM32 з використанням бібліотеки HAL (Hardware Acceleration Language), виконаний в стилі автоматного програмування і складається з безкінечного циклу та чотирьох частин:

- декларативної частини;
- функції переходів;
- функції виходів;
- функції реалізації переривань.

В декларативній частині описуються порти, декларуються константи. Кожна декларативна частина (макрос) логічно розділена коментарями, які утворюють такі підрозділи декларування як:

- опис станів переходів;
- затримки;
- опис порту та конкретних пінів вводу/виводу для світлових індикаторів;
- макроси ввімкнення або вимкнення порту вводу/виведення для світлових індикаторів (GPIO\_PIN\_SET та GPIO\_PIN\_RESET);
- вимкнення всіх світлових індикаторів (функція HAL\_GPIO\_WritePin реалізує алгоритм передачі інформації до порту вводу/виведення);
- реакція на подію відкриття шлагбаума (функція HAL\_NVIC\_EnableIRQ, аргументом якої є найменування переривання , тобто (EXTI3\_IRQn));
- макроси зчитування вхідних значень з порту вводу/виведення (зчитування даних від кнопок а допомогою функції HAL\_GPIO\_ReadPin).
- макрос синтаксису введення значення затримки у регістр МК.

## Лістинг 3.1 – Декларативна частина

```

//*****DEFINES REGION*****//
/* Визначення ідентифік. стану та його кодування цифрою в тексті програми
*/
#define SYSTEM_OFF          1
#define SYSTEM_ON           2
#define TRAIN_CROSS         3
#define LIGHT_FIRST_AND_RING 4
#define LIGHT_SECOND_AND_RING 5
#define TRAIN_CROSS_END     6

/* Визначення параметрів затримок */
#define SYSTEM_WAIT        1000
#define LIGHT_ON           1000
#define BARRIER_WORK     10000

#define LEDS_GPIOA        /* Блок (порти) керування світловими індикаторами */

/* Світлові індикатори блока */
#define LED_LIGHT_ONE      GPIO_PIN_4
#define LED_LIGHT_TWO      GPIO_PIN_5
#define RING                GPIO_PIN_6
#define BARRIER_IS_DOWN   GPIO_PIN_7 /* Сповіщувальний світловий
сигнал відкриття і закриття шлагбаума*/

#define LED_BUTTON_CROSS_END GPIO_PIN_8 /* Підсвічування кнопки закінчення
проїзду поїзда */

#define ON_GPIO_PIN_SET     /* Включення світлових індикаторів або
натискання кнопки */
#define OFF_GPIO_PIN_RESET /* Вимкнення світлових індикаторів або
віджимання кнопки */

/* Функція, яка відключає всі світлові сигнали */
#define ALL_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6
|GPIO_PIN_7|GPIO_PIN_8, GPIO_PIN_RESET);

#define CROSSING_END_ENABLE HAL_NVIC_EnableIRQ(EXTI3_IRQn);
/*Обробник події відкриття шлагбаума */
#define CROSSING_END_DISABLE HAL_NVIC_DisableIRQ(EXTI3_IRQn);
/*Обробник події заборони відкриття шлагбаума */

#define BUTTON_ON HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)
/*Ввімкнення системи ЖД переїзду */
#define BUTTON_CROSS HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)
/*Опускання шлагбаума */
#define BUTTON_CROSS_END HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_3)
/*Підняття шлагбаума*/
#define SET_DELAY TIM2->ARR /*Час роботи події*/
//*****END DEFINES REGION*****//

```

Функція переходів реалізується з використанням оператора багатопозиційного вибору `switch(state)` для кожного стану `state` реалізується

призначення нового стану `nextstate`, в залежності від наявності умов переходів.

### Лістинг 3.2 – Функція переходів (фрагмент)

```
switch(state)                /*Опис станів*/
{
    case SYSTEM_OFF:        /*Блок опису станів*/
        if(BUTTON_ON == ON) /* Натискання кнопки відповідає
                               переходу в наступний стан */
            { nextState = SYSTEM_ON      }
        else
            { nextState = SYSTEM_OFF; }
        break;

    case SYSTEM_ON:
        if (BUTTON_ON == ON)
            { if(BUTTON_CROSS == ON)
              { nextState = TRAIN_CROSS; }
              else
              { nextState = SYSTEM_ON;}
            }
        else
            { nextState = SYSTEM_OFF; }
        break;
    . . .
    default: /* Блок опису переходу автомата при збої */
        nextState = SYSTEM_OFF;
        break;
}
```

Функція виходів також реалізується за рахунок використання оператора багатопозиційного вибору `switch(state)` і для кожного стану `state` виконується призначення відповідних вихідних змінних, які, в свою чергу, ініціюють відповідні порти МК.

### Лістинг 3.3 – Функція виходів (фрагмент)

```
switch(state)                /*Опис вихідних значень/
{
    case SYSTEM_OFF:        /*Блок опису вихідного значення/
        ALL_OFF; /* Функція, яка відключає всі світлові сигнали */
        CROSSING_END_DISABLE;
        break;

    case SYSTEM_ON:
```

```

        ALL_OFF;
        CROSSING_END_DISABLE;
        break;

    case TRAIN_CROSS:                                /*a3 - L1, L2, Rg, P1*/
        HAL_GPIO_WritePin(LEDS,
            LED_LIGHT_ONE|LED_LIGHT_TWO|RING|BARRIER_IS_DOWN, ON);
        HAL_GPIO_WritePin(LEDS, LED_BUTTON_CROSS_END, OFF);
        CROSSING_END_ENABLE;
        break;

    case LIGHT_FIRST_AND_RING:                        /*a4 - L1, Rg, P1*/
        HAL_GPIO_WritePin(LEDS, LED_LIGHT_ONE|RING
            |BARRIER_IS_DOWN|LED_BUTTON_CROSS_END, ON);
        HAL_GPIO_WritePin(LEDS, LED_LIGHT_TWO, OFF);
        CROSSING_END_ENABLE;
        break;

    case LIGHT_SECOND_AND_RING:                       /*a5 - L2, Rg, P1*/
        HAL_GPIO_WritePin(LEDS, LED_LIGHT_TWO|RING
            |BARRIER_IS_DOWN|LED_BUTTON_CROSS_END, ON);
        HAL_GPIO_WritePin(LEDS, LED_LIGHT_ONE, OFF);
        CROSSING_END_ENABLE;
        break;

```

Часові параметри роботи керуючого автомату в системі управління залізничним переїздом та обробка подій реалізуються за рахунок організації переривань та функції таймера [8, 9].

#### Лістинг 3.4 – Функція обробки подій та затримок

```

void EXTI3_IRQHandler(void) /*Обробник події (переривання) закінчення
проїзду поїзда та відкриття шлагбауму*/
{
    /* USER CODE BEGIN EXTI3_IRQn 0 */
    /*Умова дає затримку, чого достатньо для повного підтвердження того, що був
відсутній дребезг*/
    if(BUTTON_CROSS == OFF && BUTTON_CROSS_END == OFF)
    {
        state = TRAIN_CROSS_END;
    /*Моментальний перехід у стан відкриття шлагбауму*/
        DelayMs = BARRIER_WORK;
        SET_DELAY = DelayMs-1; /*Безпосередня зміна часу виклику*/
    }

    /* USER CODE END EXTI3_IRQn 0 */
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_3);
    /* USER CODE BEGIN EXTI3_IRQn 1 */
    /* USER CODE END EXTI3_IRQn 1 */
}

```

```

void TIM2_IRQHandler(void) /*Визначення часу роботи наступного стану
перериванням від таймера після закінчення часу роботи поточного стану*/
{
    /* USER CODE BEGIN TIM2_IRQn 0 */
    state = nextState;
    if (state == SYSTEM_OFF || state == SYSTEM_ON)
    { DelayMs = SYSTEM_WAIT; }
    else if (state == LIGHT_FIRST_AND_RING || state ==
LIGHT_SECOND_AND_RING)
    { DelayMs = LIGHT_ON; }
    else if (state == TRAIN_CROSS || state == TRAIN_CROSS_END)
    { DelayMs = BARRIER_WORK; }
    else
    { DelayMs = SYSTEM_WAIT; }

    /* USER CODE END TIM2_IRQn 0 */
    HAL_TIM_IRQHandler(&htim2);
    /* USER CODE BEGIN TIM2_IRQn 1 */
    SET_DELAY = DelayMs-1; /* Безпосередня зміна часу виклику події
переповнення таймера*/
    /* USER CODE END TIM2_IRQn 1 */
}

```

Програмний код макету пристрою керування на основі МК STM32F030C8T6 наведений у додатку В.

Для моделювання електричної принципової схеми пристрою керування залізничним переїздом використаємо інструментальні засоби САПР Proteus . Мікросхеми STM32F030C8T6 в бібліотеках наявної версії Proteus немає, тому для моделювання розробленої схеми будемо використовувати одну з мікросхем сімейства Intel MCS 51, а саме AT89C52. Програма реалізації пристрою керування залізничним переїздом на базі мікроконтролера MCS 51 виконана на мові програмування C у середовищі розробки Keil.

При включенні (state = SYSTEM\_ON) система знаходиться в режимі очікування взаємодії датчика (імітація кнопкою TRAIN CROSSING START) з поїздом (рис. 3.4). Час очікування дорівнює 1 секунді (DelayMs = SYSTEM\_WAIT), чого достатньо для швидкої реакції контакту рухомого складу з датчиком. Якщо система не виявила взаємодію або контакт, то цикл очікування повторюється. За цю операцію відповідає обробник події (переривання) на подію закінчення часу (Timer0\_isr).

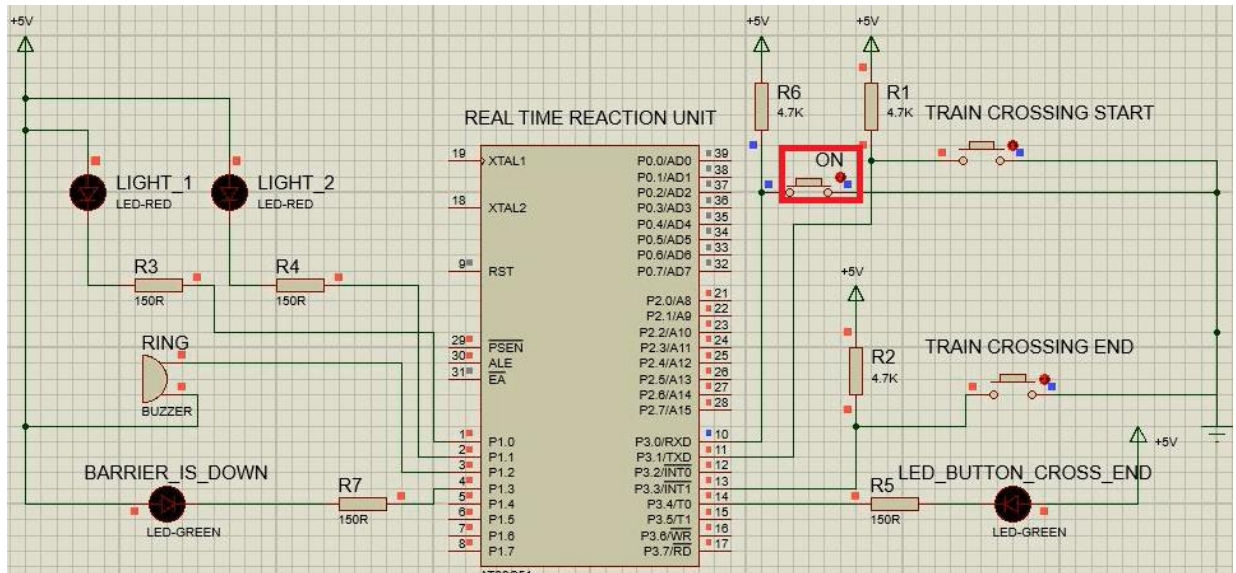


Рисунок 3.4 – Старт системи

При взаємодії датчика та поїзда (затиснута кнопка TRAIN CROSSING START), система вмикає залізничний світлофор, сирену та закриття шлагбаума (state = TRAIN\_CROSS). На рис.3.5 робота зі шлагбаумом відображена за допомогою світлодіода зеленого кольору BARRIER\_IS\_DOWN, який відповідає за закриття, відкриття шлагбаума та очікування проїзду поїзда (шлагбаум перекриває проїзд дорожнього транспорту через залізничну колію на певний проміжок часу).

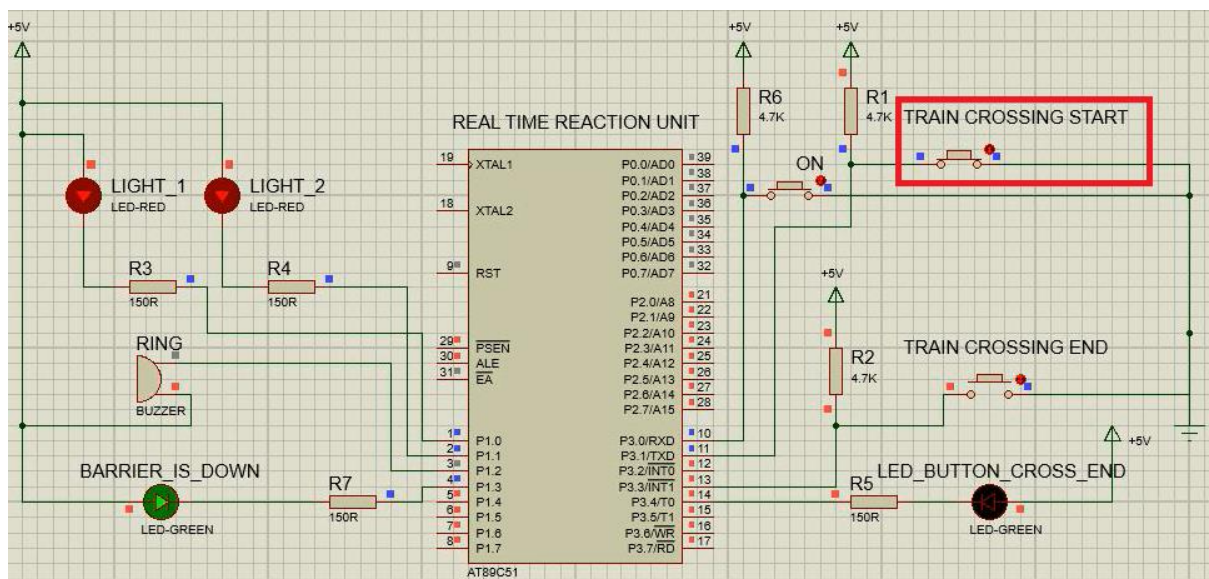


Рисунок 3.5 – Взаємодія датчика з поїздом

На рис. 3.6 та рис. 3.7 відображено миготіння червоних світлодіодів, які імітують залізничний світлофор. Рисунки відображають роботу гіперавтомату, який контролює світлофор та відображає, за допомогою зеленого світлодіода LED\_BUTTON\_CROSS\_END, взаємодію поїзда із датчиком, який знаходиться поза залізничним переїздом (імітація кнопкою TRAIN CROSSING END).

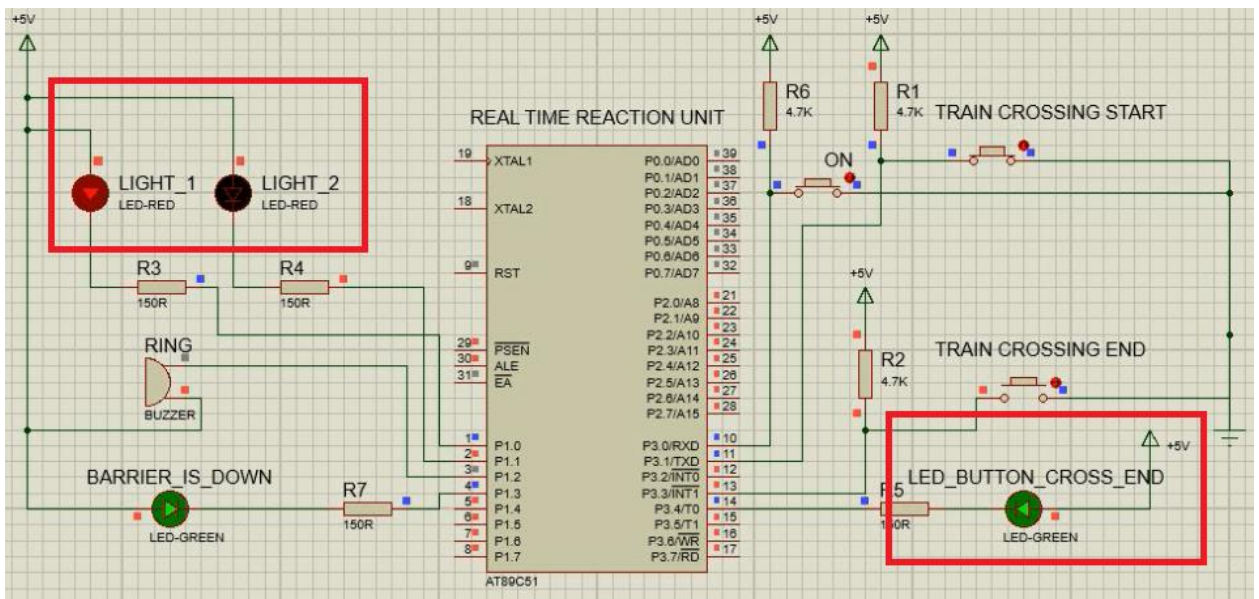


Рисунок 3.6 – Стан системи LIGHT\_FIRST\_AND\_RING

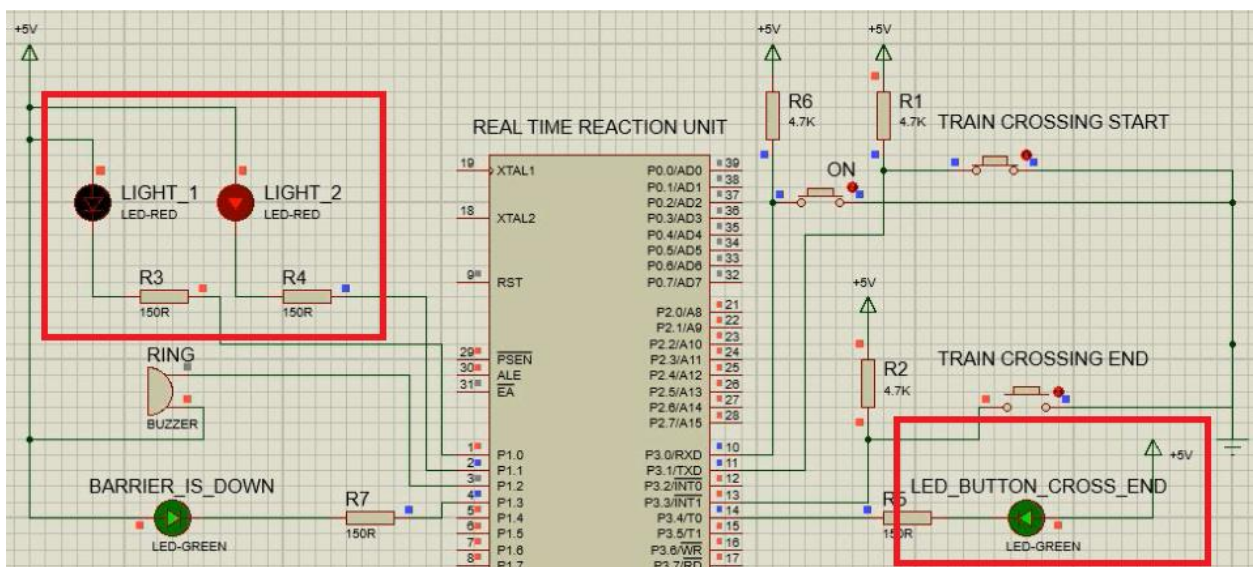


Рисунок 3.7 – Стан системи LIGHT\_SECOND\_AND\_RING

Для початку реакції системи на те, що поїзд покинув залізничний переїзд, на вимкнення світлофора та сирени та на відкриття шлагбаума для проїзду дорожнього транспорту (state = TRAIN\_CROSS\_END), поїзду необхідно почати взаємодію з другим датчиком, який знаходиться за залізничним переїздом (TRAIN CROSS END) та завершити її з першим датчиком (TRAIN CROSSING START). Імітація завершення контакту зображена на рис. 3.8. Проте, варто зауважити, що система до сих пір знаходиться у циклі гіперавтомату.

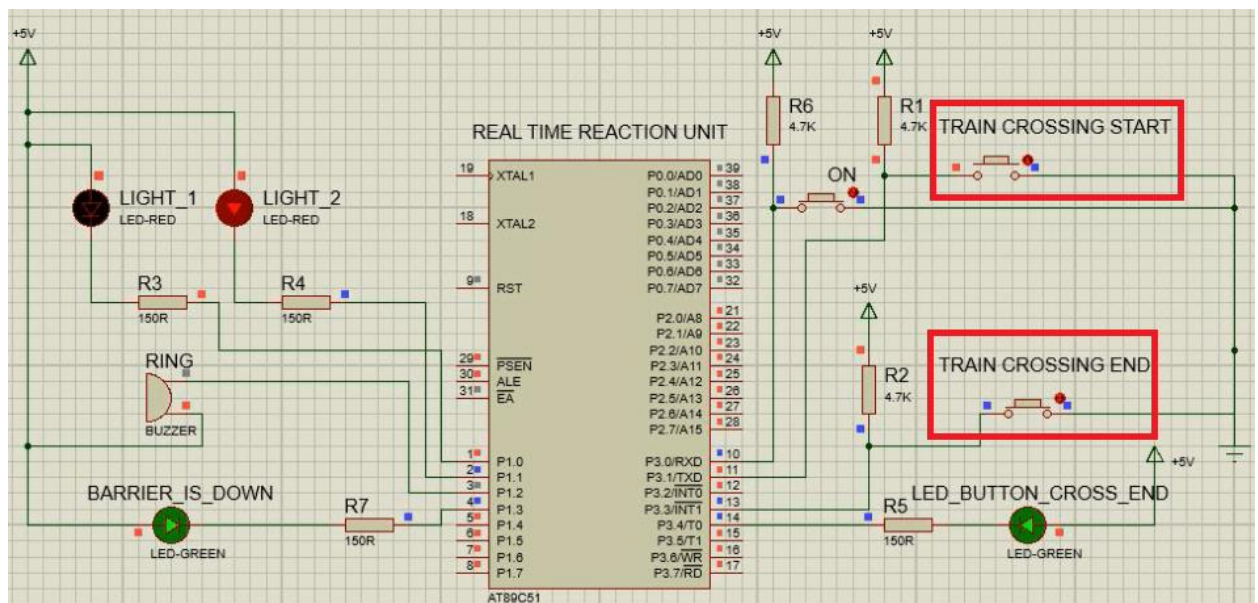


Рисунок 3.8 – Початок проїзду поїзда залізничного переїзду.

Єдиним виходом з циклу гіперавтомату системи є подія повного проїзду (поїзд повністю покинув залізничний переїзд) – завершення взаємодії поїзда з датчиками перед та за залізничним переїздом (ButtonCrossEndHandler). Варто зауважити, що обробка події повинна бути після відтискання імітаційної кнопки другого датчика. Коли поїзд перестав взаємодіяти з датчиком поза межею переїзду, відбувається обробка цієї події, яка автоматично переводить систему у наступний стан (state = TRAIN\_CROSS\_END) з вимкненням світлофора та сирени (рис. 3.9).

Далі відбувається підняття шлагбаума. Після того, як світлодіод, імітуючий роботу шлагбаума перестав горіти, дорожній автотранспорт може продовжити рух. В цей час, якщо система увімкнена (натиснута кнопка ON), то система повертається у стан очікування наступного проїзду поїзда (state = SYSTEM\_ON). Інакше – система вимкнена (state = SYSTEM\_OFF).

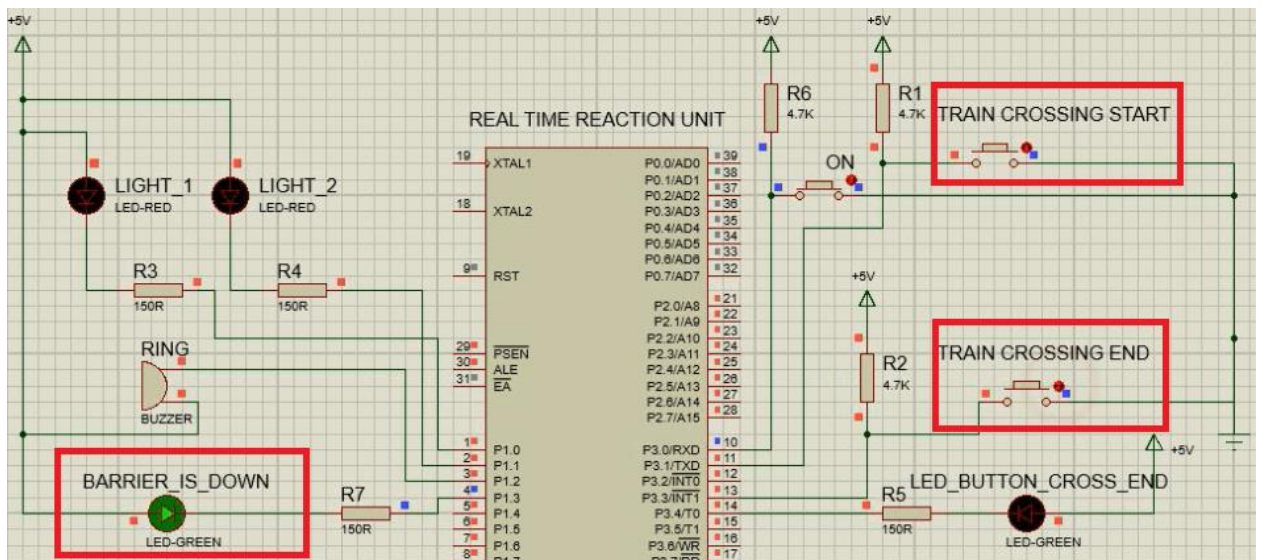


Рисунок 3.9 – Завершення проїзду поїзда (state = TRAIN\_CROSS\_END)

Повна програма імітації реалізації пристрою керування залізничним переїздом на базі МК Intel MCS 51 на мові програмування C наведена у додатку Б.

### 3.3 Макетування пристрою керування залізничним переїздом

Для макетування запропонованого пристрою керування залізничним переїздом був розроблений макет пристрою на основі мікроконтролера STM32F030. Для мікроконтролера STM32F030 напруга датчиків не повинна перевищувати 3.3В;

Для макетування системи використані світлодіоди для підсвічування імітаційної кнопки завершення проїзду поїзда та дозволу відкриття

шлагбаума, а також імітації роботи шлагбаума (закриття та його відкриття) були використані світлодіоди типу FYL-3004GD. В якості червоного світла для залізничного світлофору використаний світлодіод C5SMF-RJS-CU14QBV1. Для більш чіткого відображення червоного світла на фоні підсвічування кнопки та імітації роботи шлагбаума необхідно вибрати світлодіод з більшою яскравістю. В залежності від напруги живлення (2.8-3.3В), червоний світлодіод має яскравість у 1520-2564 кд/м. Для звукової сирени був використаний електромагнітний випромінювач HCM1203X з робочою напругою у 3В, потужністю випромінювання у 85дБ та робочою частотою у 2.3 кГц. На рис.3.11 показано макет системи залізничного переїзду у режимі готовності. Система ввімкнена (виділена синім прямокутником на рис.3.10), проте без проїзду поїзда першого датчика руху (виділена синім прямокутником на рис.3.6) система не спрацює і буде знаходитись в режимі очікування нескінченно, доки система не буде вимкнена взагалі.

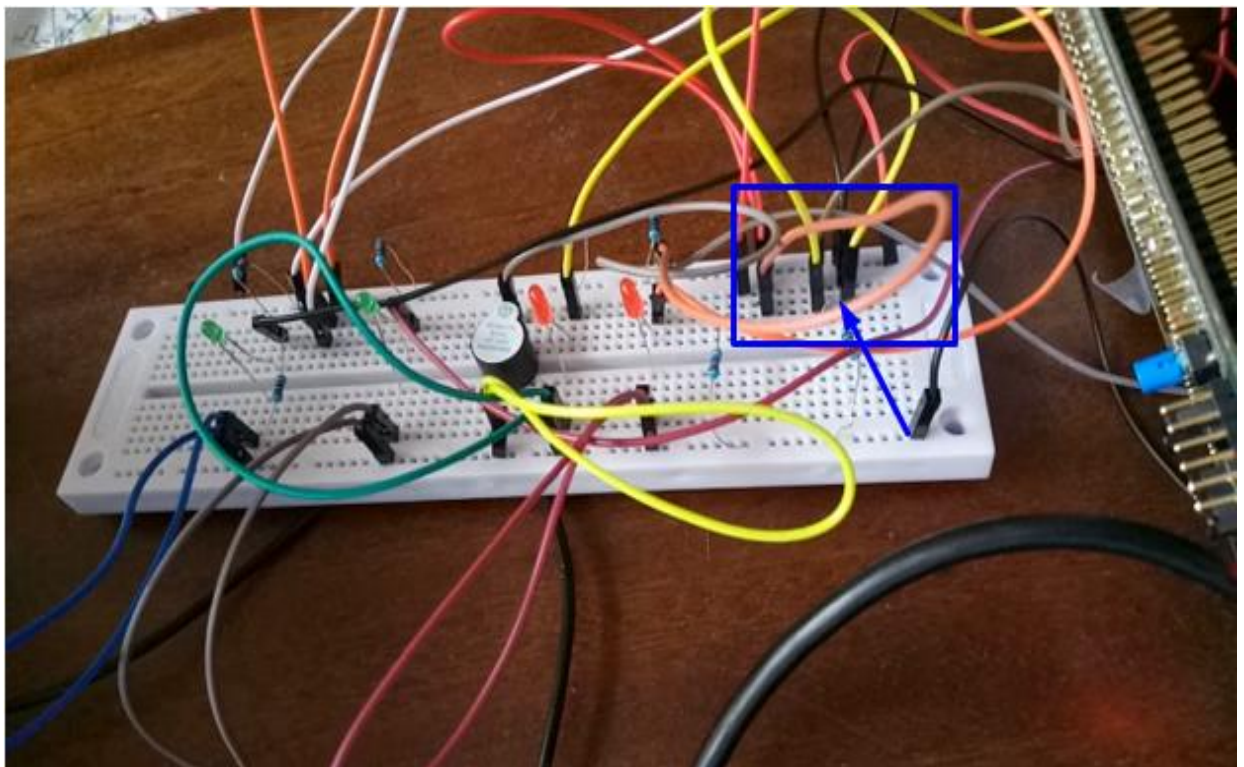


Рисунок 3.10 – Макет системи в режимі очікування

На рис.3.11 показано включення системи в режим заборони переїзду (початок проїзду поїзда). Слід зауважити, що система працює певний проміжок часу, вказаний у технічних налаштуваннях (код).

Повний цикл роботи світлофора (проїзд поїзда) відображений на рис 3.11, рис.3.12 та рис.3.13, який має неперервний процес миготіння червоних світлодіодів з подальшим звуковим супроводом сирени. Увімкнений зелений світлодіод імітує роботу шлагбаума (закриття, заборона пересічення транспорту, відкриття шлагбаума). Слід зауважити, що перетинання поїздом першого датчика руху (датчика наближення, який виділений синім прямокутником на рис.3.13) також відображене на рис. 3.11, рис.3.12 та рис.3.13. Рис.3.12 відображає дозвіл на вимкнення системи очікування проїзду поїзда та повернення системи у стан очікування наступного проїзду за допомогою підсвічування імітації датчика кнопкою (на рис.3.13 зелений світлодіод виділений синім прямокутником).

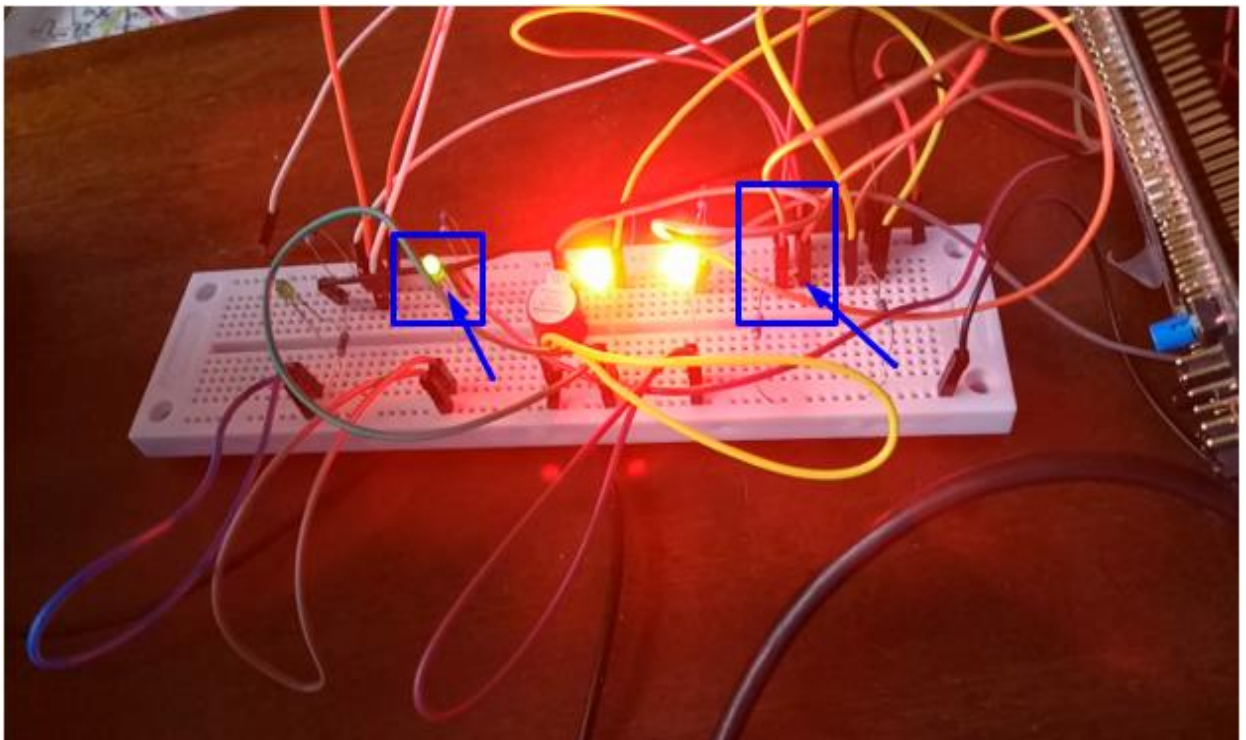


Рисунок 3.11 – Включення системи проїзду поїзда (ввімкнення світлофору, звукового сигналу та закриття шлагбаума)

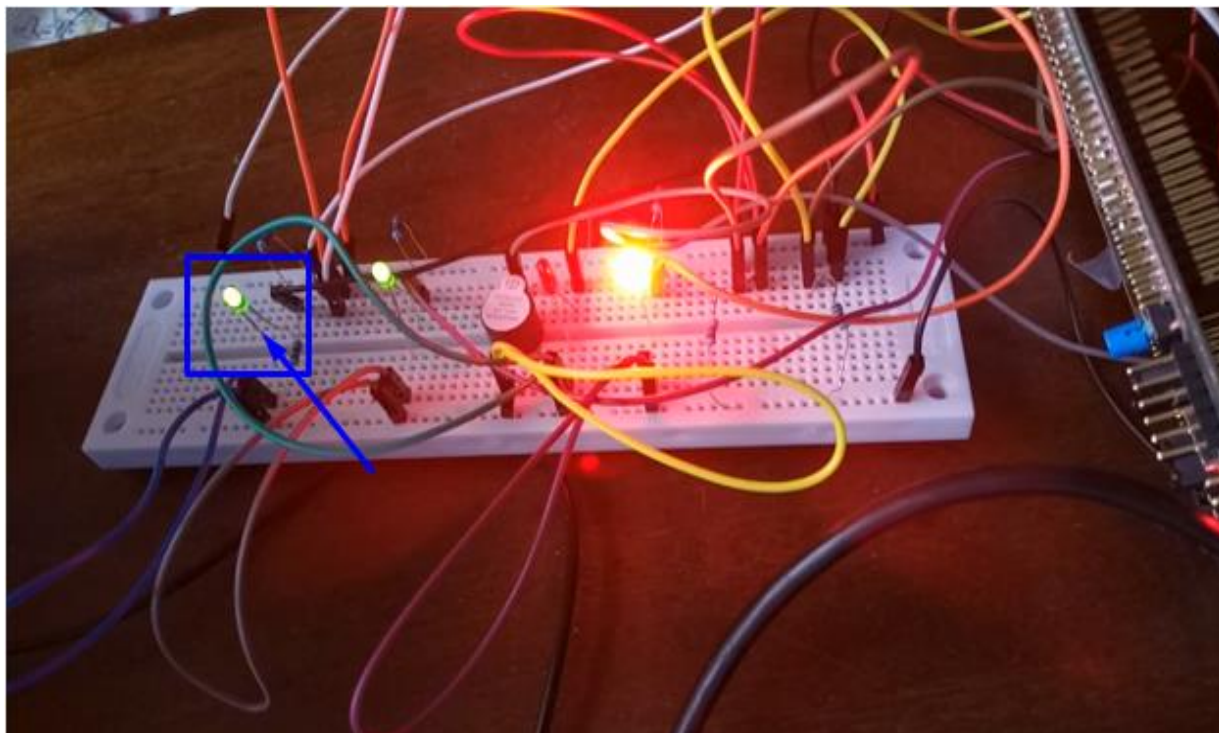


Рисунок 3.12 – Зміна стану світлофора та дозвіл на вимикання системи

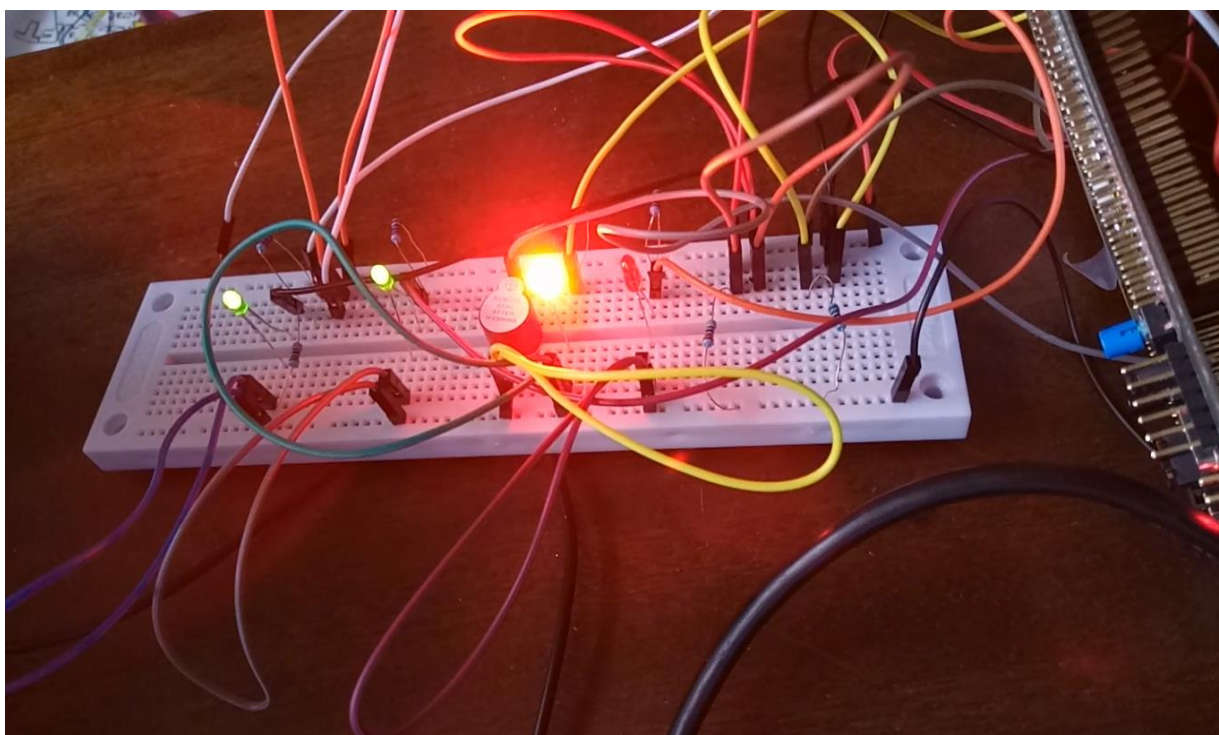


Рисунок 3.13 – Миготіння світлофора (почергове увімкнення червоних світлодіодів)

На рис.3.14 відбувається початок дії для переходу системи у стан очікування. Необхідно зімітувати пересічення датчика поїздом. Система повинна перейти у стан очікування тільки якщо зімітовані датчики вимкнені. Спочатку потрібно увімкнути датчик, який знаходиться поза переїздом (друга взаємодія датчика і поїзда), потім, після проїзду останнього вагона, вимкнути перший датчик, з яким почав контактувати поїзд (перша взаємодія датчика та поїзда, яка виникає на рис.3.6) та вимкнути датчик, який взаємодія з останнім вагоном після його взаємодії з першим датчиком. Повний цикл відображено на рис3.14 – рис.3.16. Ключові моменти на кожному з рисунків відображені синім прямокутником.

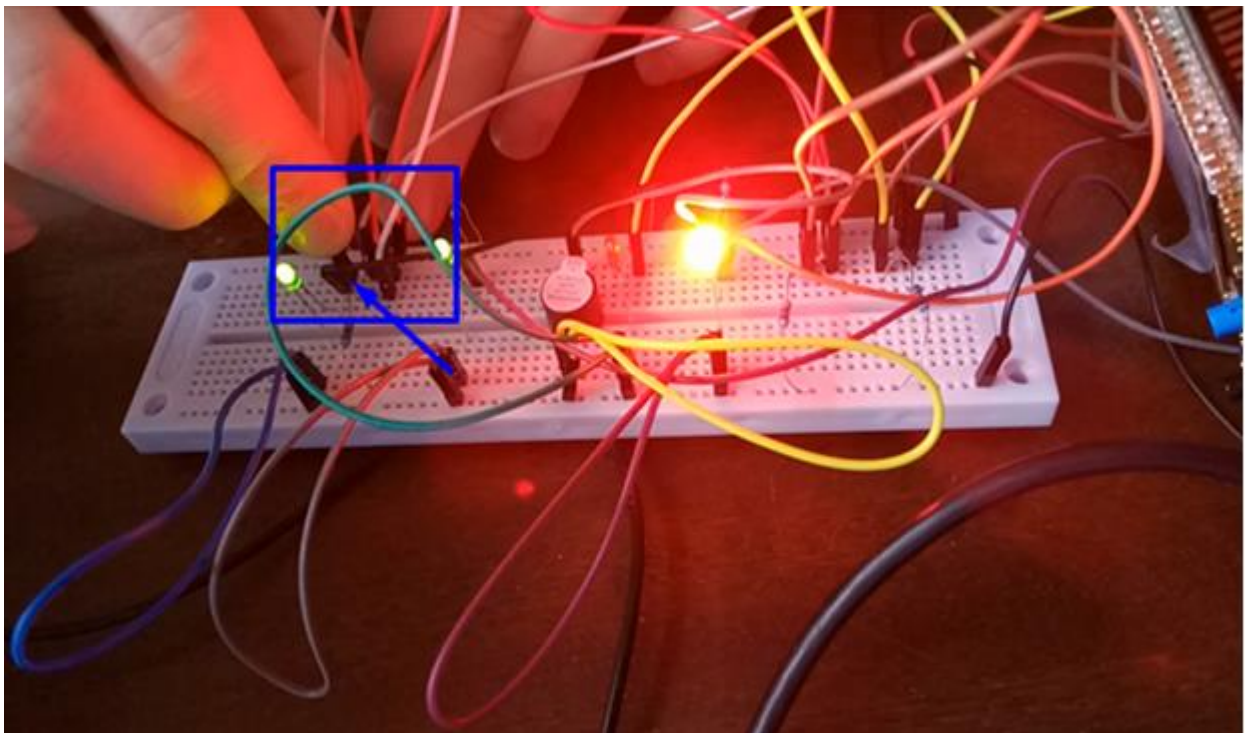


Рисунок 3.14 – Другий датчик почав взаємодіяти з поїздом

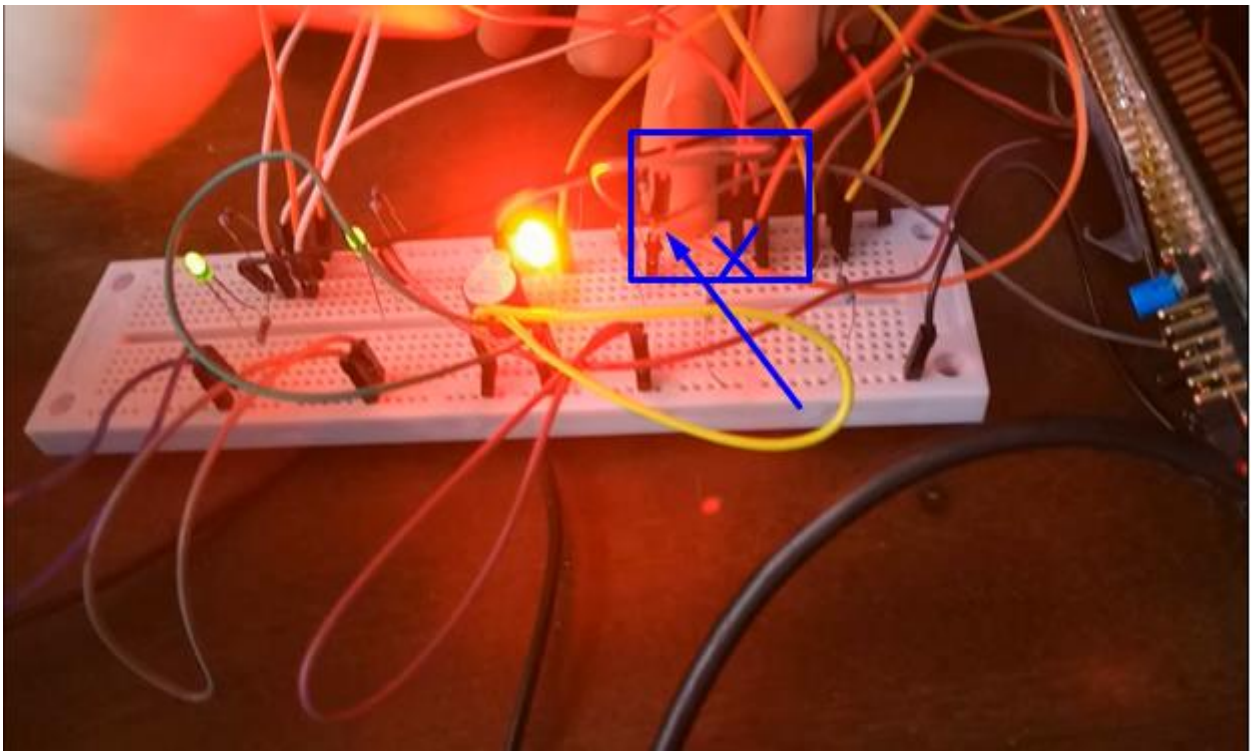


Рисунок 3.15 – Перший датчик перестав взаємодіяти з поїздом (проїзд останнього вагона)

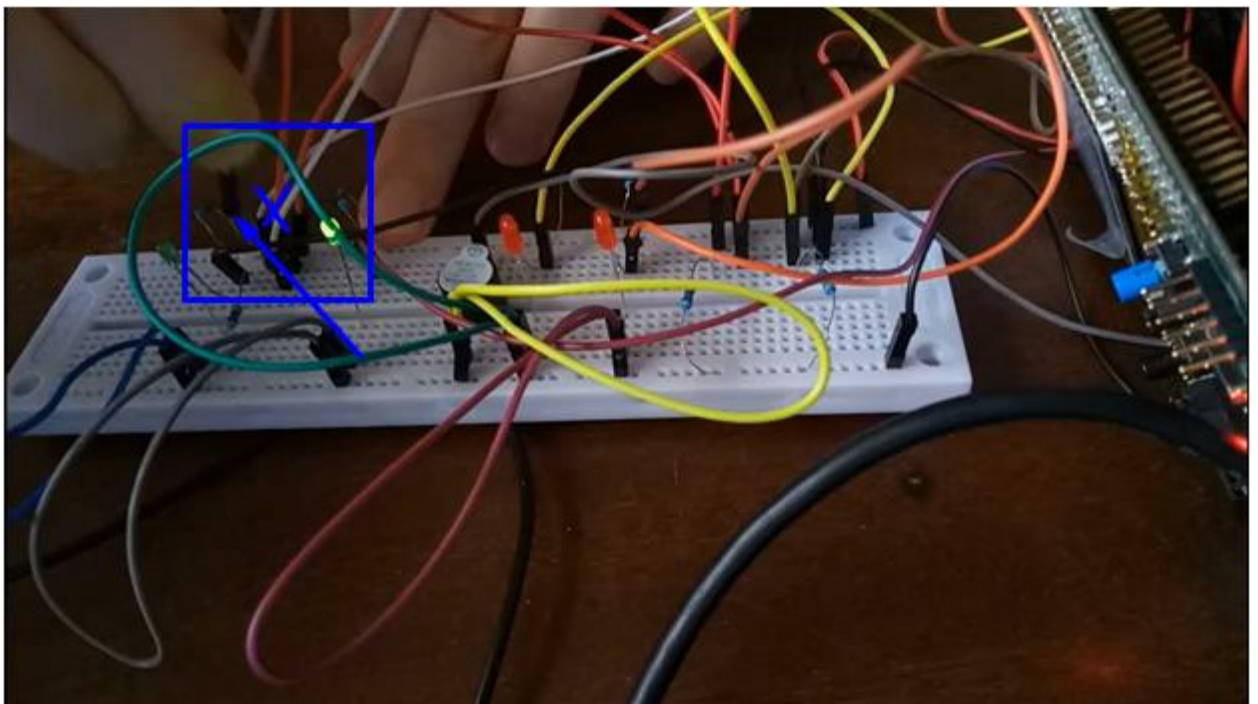


Рисунок 3.16 – Другий датчик перестав взаємодіяти з поїздом (поїзд покинув переїзд)

На рис. 3.16 горить зелений світлодіод, який імітує відкриття шлагбаума для відновлення проїзду автотранспорту.

Далі система вертається у початковий стан та чекає на нову взаємодію датчика та поїзда. Також, слід зауважити, що система не вимкнена, що демонструє увімкнута кнопка запуску системи, виділена синім прямокутником. Початковий стан системи представлений на рис.3.10.

## ВИСНОВКИ

В ході виконання бакалаврської кваліфікаційної роботи розроблено мікроконтролерний пристрій керування автоматичним залізничним переїздом на основі кінцевого керуючого автомата. Ця розробка є виконана в напрямку впровадження в систему залізничної автоматики сучасних мікроконтролерних систем на заміну релейно-контактним схемам.

При побудові пристрою керування використана модель часового автомата Мура з надбудовою у вигляді гіперавтомата. Це дозволило спростити модель автомата в режимі зациклювання та обробляти події від зовнішніх датчиків. Реалізована окрема процедура обробки події – сигналу датчика наближення поїзду до залізничного переїзду. Розроблені принципи організації зворотного зв'язку пристрою керування з виконавчими пристроями залізничного переїзду.

Керуючий автомат реалізований на базі мікроконтролера сімейства STM32 (STM32F030C8T6). Програмна реалізація алгоритму керування здійснена на мові програмування C у середовищі розробки Keil. Моделювання розробленого пристрою керування проводилося інструментальними засобами САПР Proteus. Для моделювання використовувалася бібліотека з мікроконтролерами Intel MCS 51. Макетування здійснювалося на налагоджувальній платі STM32 з використанням кольорових світлодіодів та звукового генератора.

Результати моделювання та макетування підтвердили працездатність та ефективність розробленого мікроконтролерного пристрою керування автоматичним залізничним переїздом.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Національний стандарт України. ДСТУ 4178-2003. Комплекси технічних засобів систем керування та регулювання руху поїздів. Функційна безпечність і надійність. Вимоги та методи випробування : чинний від 01.07.2003 р. – К.: Держспоживстандарт України, 2003. – 32 с.
2. Інструкція з улаштування та експлуатації залізничних переїздів [Електронний ресурс] / Законодавство України : Документ z0162-07. – Редакція від 15.04.2011, підстава – z0439-11. – Режим доступу: [www / URL: https://zakon.rada.gov.ua/laws/show/z0162-07](http://www.zakon.rada.gov.ua/laws/show/z0162-07) – 25.04.2019 г.
3. Шкіль А.С., Автоматизированное проектирование систем логического управления с использованием шаблонов автоматного программирования / А.С. Шкіль, Э.Н. Кулак, И.В. Филиппенко, М.В. Гога, Д.Е. Кучеренко. // *Радіоелектроніка та інформатика* – 2018. – №3. – С. 75-81.
4. Shkil A.S. Design timed FSM with VHDL Moore pattern / М.А. Miroshnyk, А.С. Shkil, E.N. Kulak, D.Y. Rakhlis, А.М. Miroshnyk, N.V. Malahov // *Radio Electronics, Computer Science, Control*. – 2020. – №2(53). – P. 137-148. DOI: <https://doi.org/10.15588/1607-3274-2020-2-14>.
5. Шкіль О.С. Мікроконтролерна система керування автоматичним залізничним переїздом / О.С. Шкіль, І.В. Філіппенко, М.В. Гога // *Інформаційно-керуючі системи на залізничному транспорті*. – 2018. – № 4. – С. 81-82.
6. Insider's Guide to the STM32 ARM®Based Microcontroller [Електронний ресурс] / Hitex development tools. – Режим доступу: [www / URL: https:// www.hitex.com](http://www.hitex.com) – 20.04.2025 г.
7. Проектування та аналіз електричних схем в програмному середовищі Proteus VSM. Методичні вказівки до самостійної роботи по курсу "Проектування мікропроцесорних систем керування технологічними процесами". Медвідь В.Р., Пісцью В.П. – Тернопіль: ТНТУ, 2018 – 26 с.

8. Матюшин А.О. Программирование микроконтроллеров: Стратегия и тактика / А.О. Матюшин. – М.: ДМК Пресс, 2017. – 356 с.

9. Shkil O.S. Temporal events processing models in finite state machines / М.А. Miroschnyk, O.S. Shkil, S.I. Shmatkov, А.М. Miroschnyk, К.У. Pshenychnyi // Radio Electronics, Computer Science, Control. – 2023. – №4 (67). – P. 49-57.  
<https://doi.org/10.15588/1607-3274-2023-4-5>.

