МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ

ТЕХНОЛОГИЙ УПРАВЛІННЯ

Лабораторна робота № 3

з курсу «Моделювання складних систем»

«Побудова дискретно-подієвих моделей у середовищі AnyLogic»

Виконав:

студент групи КН 36-б

Чуркін Р.Д.

Перевірила:

ст. викл. каф. ПІІТУ

Єршова С. І.

ХАРКІВ 2019

**Мета виконання лабораторної роботи**

Метою виконання роботи є:

1) навчитися створювати нові класи активних об’єктів;

2) навчитися працювати з портами, подіями та повідомленнями;

3) навчитися розробляти діаграмами станів, створювати ієрархічні стани та реалізовувати обмін повідомленнями між діаграмами.

**Завдання на виконання**

1) ознайомитися з об’єктами моделювання;

2) провести моделювання функціонування об’єктів;

3) розробити презентації імітаційних моделей лічильника та пішохідного переходу;

4) проаналізувати результати моделювання;

5) розробити звіт до лабораторної роботи, який повинен містити короткий опис ходу роботи та результати експериментів з імітаційними моделями:

– десяткового лічильника;

– пішохідного переходу;

6) виконати індивідуальне завдання відповідно до варіанту.

Індивідуальне завдання.

Змінити модель таким чином, щоб генератор створював «тики» в випадкові моменти часу; змінити модель таким чином, щоб частота миготіння зеленого світла регулювалася за допомогою слайдера; змінити модель автоматичного світлофора автомобілів і пішоходів (без кнопки), видаливши переходи по сигналу і підберіть відповідні таймаути для узгодженої роботи світлофорів.

**Хід виконання роботи**

1 Розробка дискретно-подієвої моделі десяткового лічильника.

Згідно з методичними вказівками було створено модель з назвою «Model», та створено два пов’язаних між собою активних класів об’єктів «Gen» та «Counter». Клас «Gen» відповідає за генерацію імпульсів та передачу цих імпульсів до лічильника. Структура класу «Gen» представлена на рисунку 1.

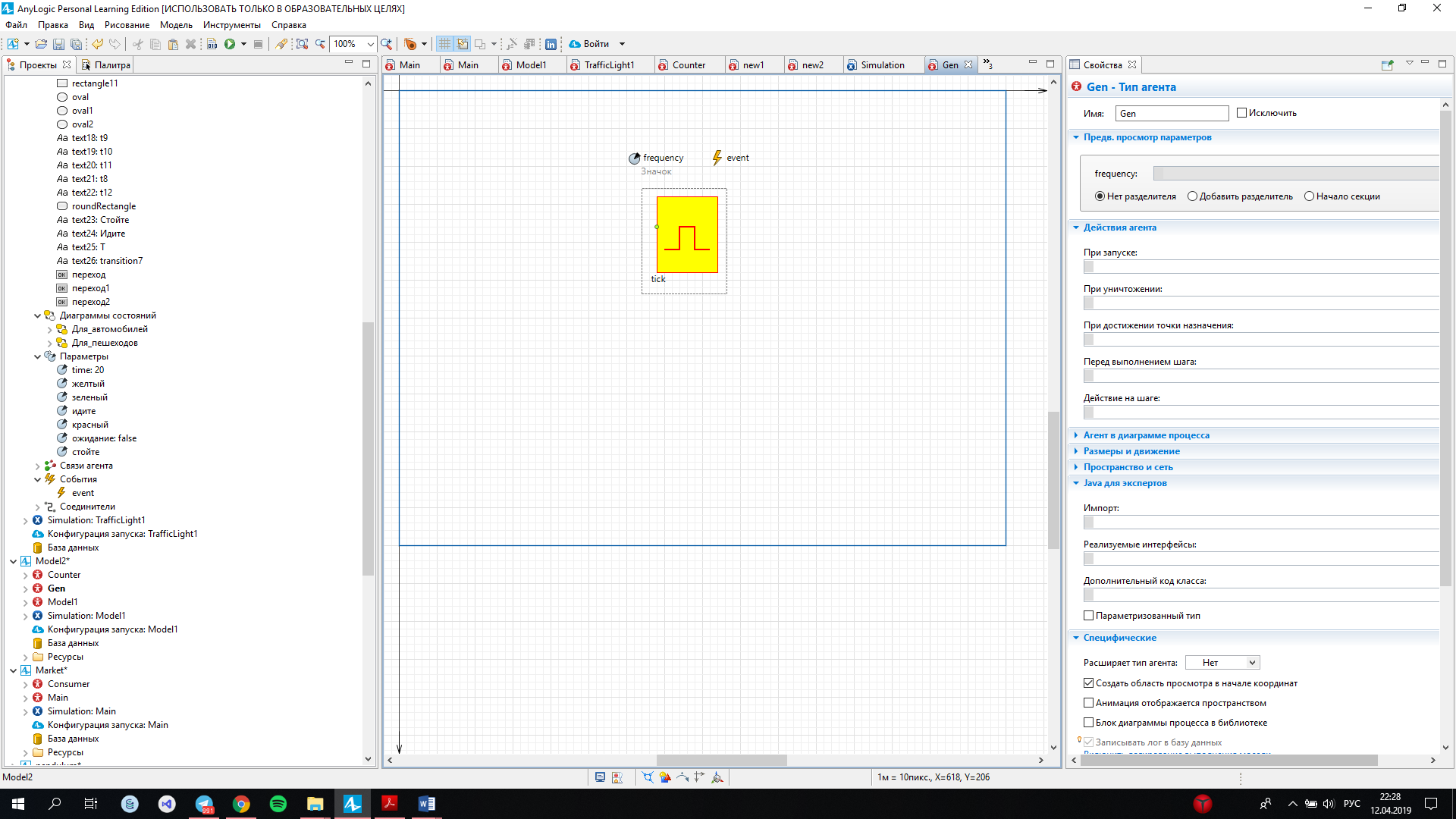


Рисунок 1 — Структура класу «Gen»

Клас «Counter» призначений для моделювання розряду десяткового лічильника. Структура класу «Counter» представлена на рисунку 2.

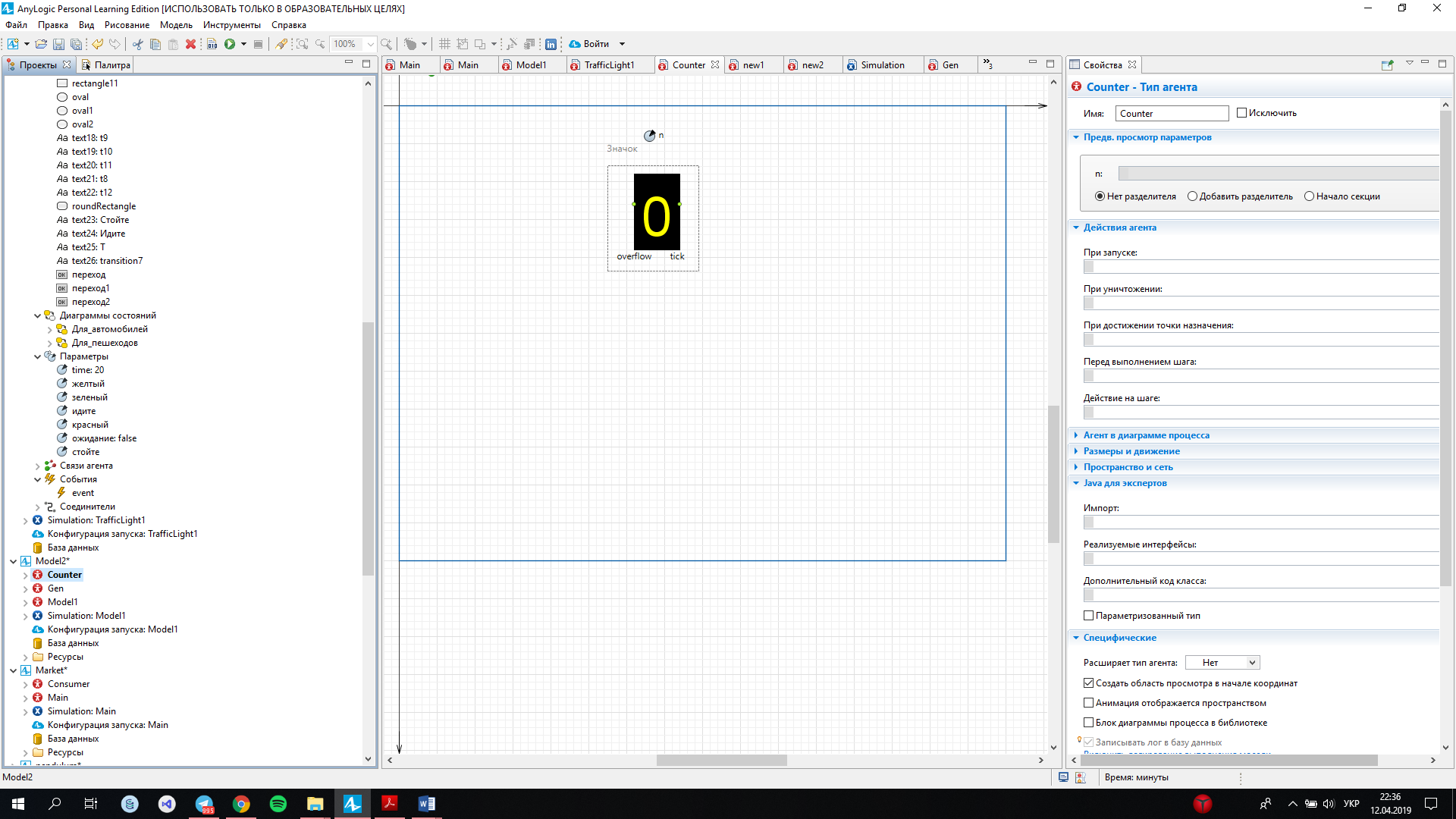


Рисунок 2 – Структура класу«Counter»

При додаванні елементу порт під назвою «tick» необхідно у властивостях «Дії обробки повідомлень» додати код, представлений на рисунку 3.

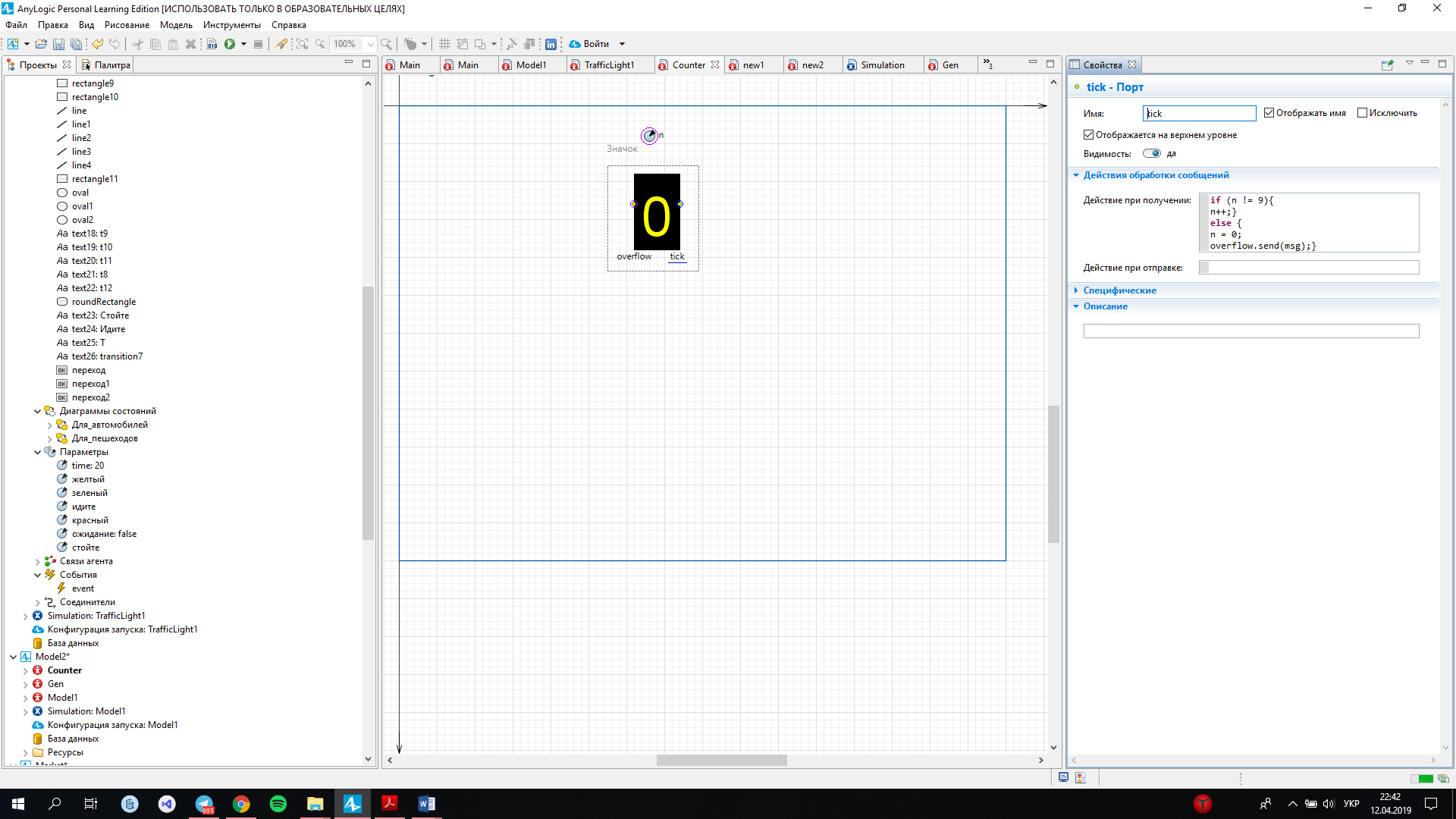


Рисунок 3 — Налаштування лічильника

Цей код вказує на те, що значення числа буде збільшуватися, доки не дійде до 9, потім лічильник обнуляється та відправляє повідомлення до наступного лічильника.

Головна форма моделі буде містити 3 лічильники та один генератор, які необхідно правильно поєднати між собою. В результаті отримаємо готову модель десяткового лічильника (рисунок 4).

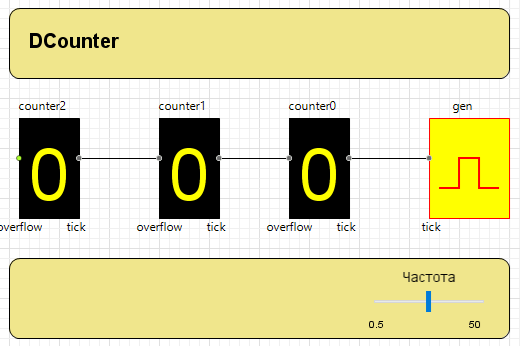


Рисунок 4 — Готова модель десяткового лічильника

Проведені експерименти.

Встановимо час виконання моделі t=20 сек та перевіримо, які значення чисел на лічильниках вдасться отримати при зміні параметру «частота». Отримані результати занесені у таблицю 1.

Таблиця 1 — Результати експерименту

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Частота | 0 | 1 | 2 | 3 | 4 |
| Отримане значення | 1 | 21 | 41 | 61 | 81 |

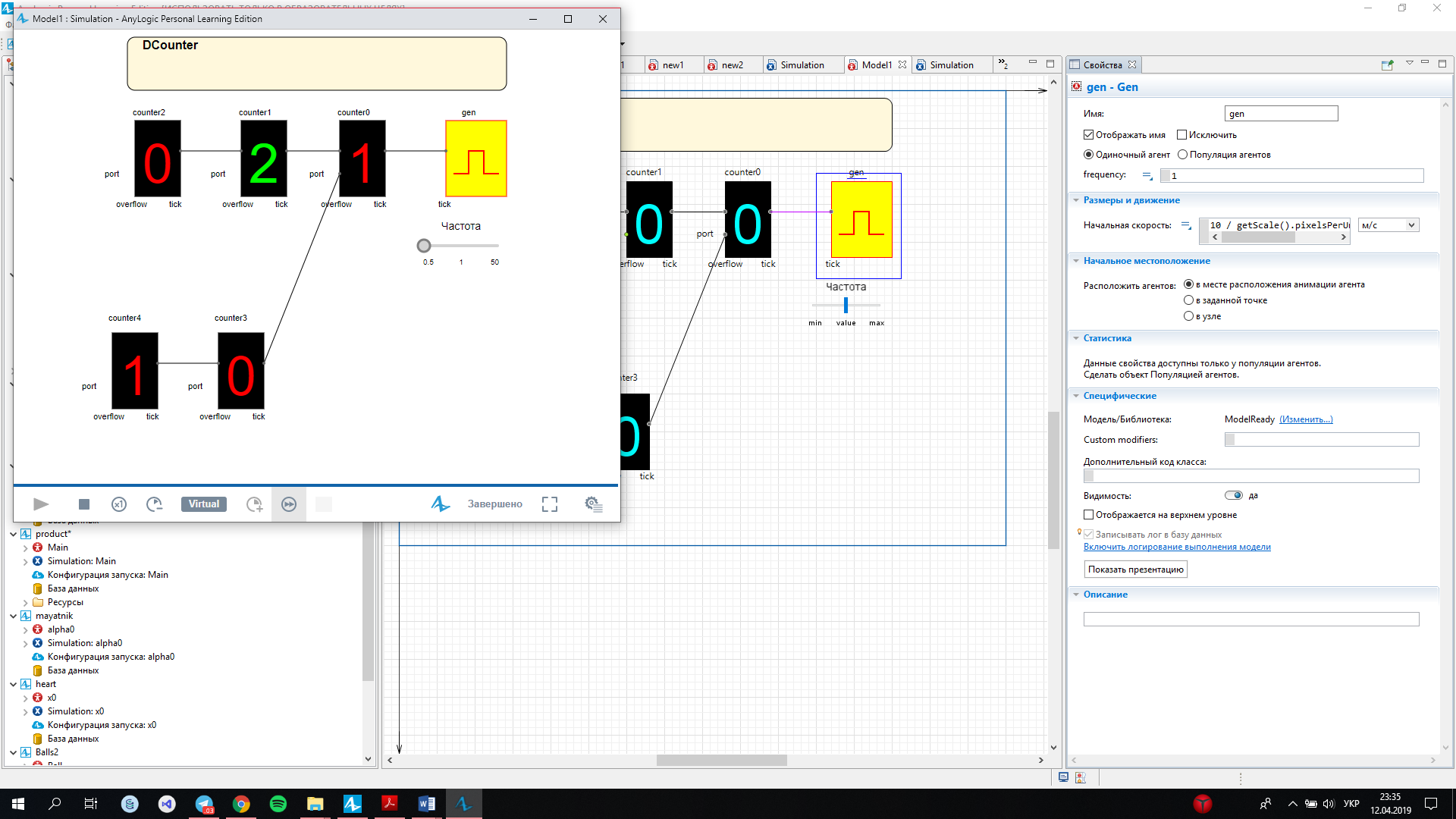


Рисунок 5 — Отримані результати при частоті рівній 1

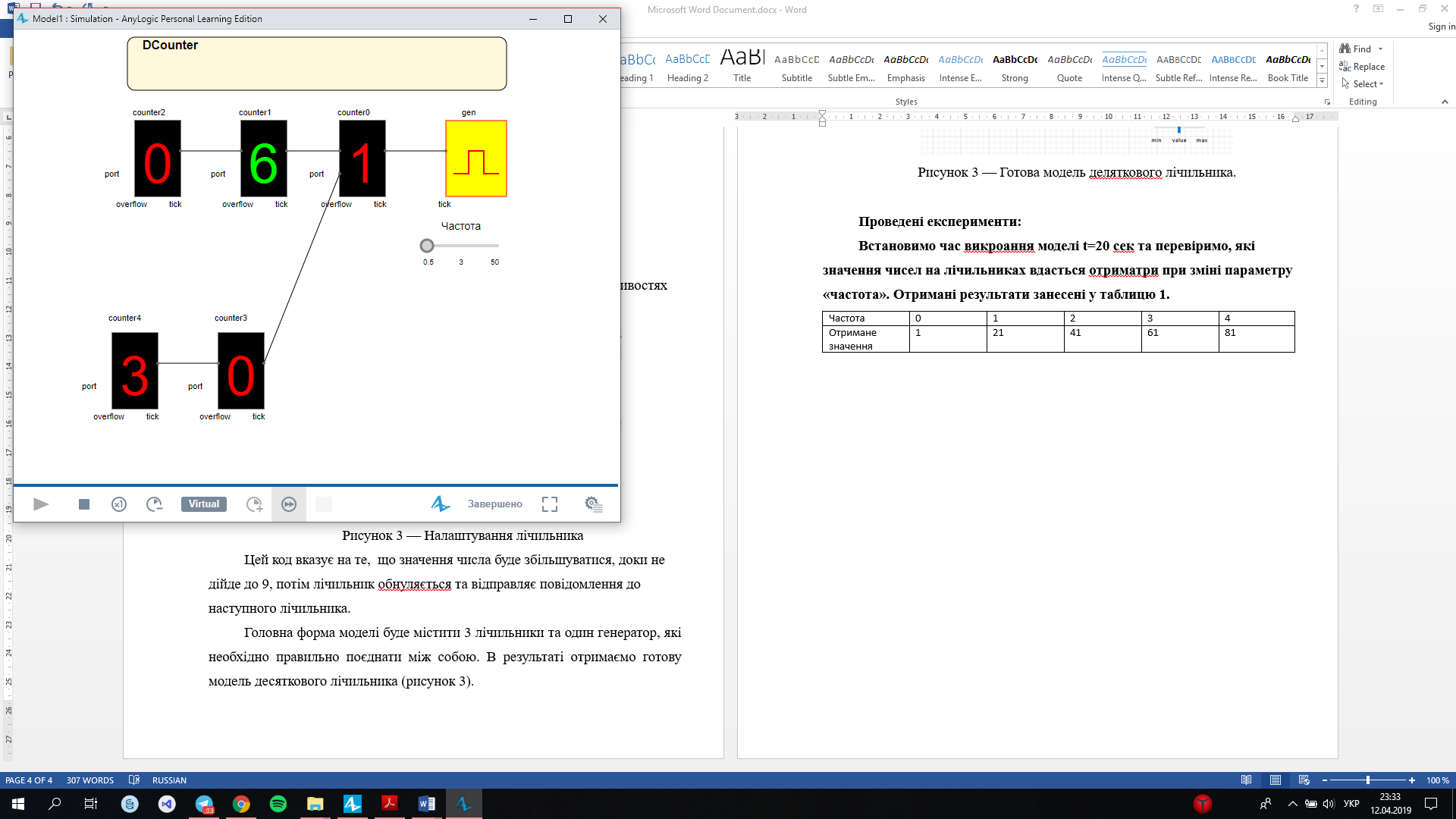


Рисунок 6 — Отримані результати при частоті рівній 3

Таким чином можна зробити висновок, що при збільшенні частоти збільшується швидкість генерації імпульсів і тим швидше змінюються числа на лічильнику. Навпаки, при зменшенні частоти генерація чисел буде сповільнюватися.

2 Розробка дискретно-подієвої моделі пішохідного переходу.

Модель буде представляти собою систему станів, що пістійно змінюються. Для реалізації моделі необхідно розробити діграму станів для автомобілів та пішоходів. Кінцева діаграма станів для автомобілів представлена на рисунку 7.

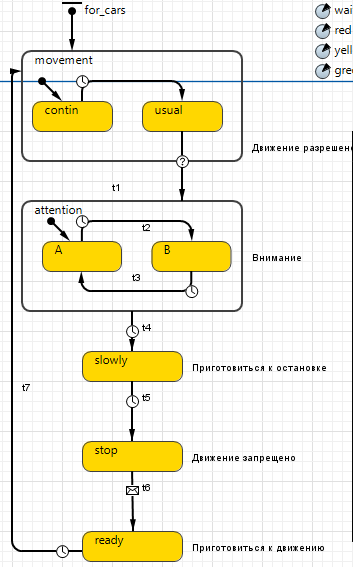


Рисунок 7 — Діаграма станів для автомобілів

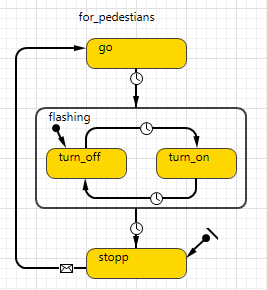


Рисунок 8 — Діаграма станів для пішоходів

Були встановлені необхідні налаштування згідно з методичними вказівками та розроблено презентацію моделі.

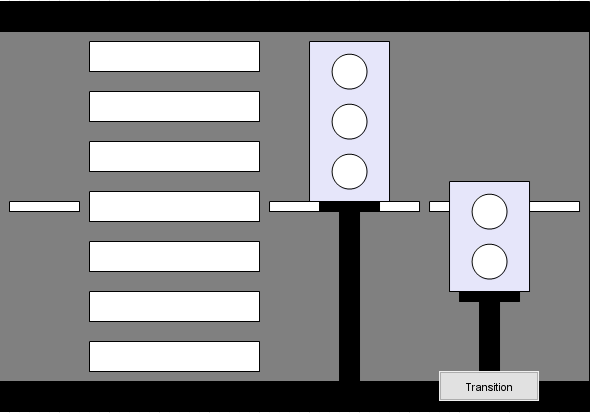


Рисунок 9 — Презентація розробленої моделі

Створена модель була запущена та перевірена на працездатність.

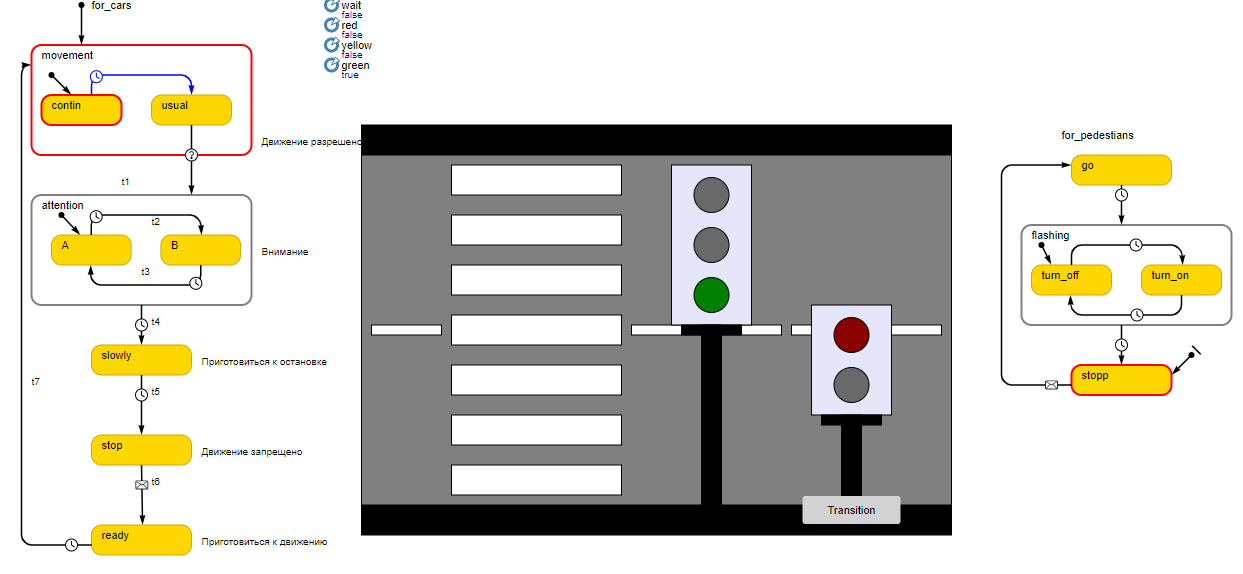


Рисунок 10 – Результат запуску моделі

Проведені експерименти.

Здійснимо перехід від початкового стану до останнього змінюючи таймаути переходів та поспостерігаємо за змінами.

1. Перехід t12 відбувається по таймауту. За замовченням це значення 5. Цей перехід відповідає за час перевірки статусу очікування. Тобто кожні 5 секунд відбувається перевірча на те, чи натиснута кнопака «перехід». Збільшивши значення отримаємо довший проміжок часу на перевірку, що змусить пішоходів чекати на зелене світло довше.
2. У разі, якщо кнопка була натиснута, система переходить у інший стан. Починає блимати зелене світло водіям, що говорить про те, що зелене світло незабаром закінчиться. Переходи t2 та t3 поєднують 2 стани, а саме горить чи не горить зелене світло. За замовченням стоять значення 1. Тобто одну секунду зелене світло горить, а потім одно секунду не горить. Збільшивши час, ормимаємо меньшу кількість блимків зеленого кольору. Цей процес відбувається відповідно до значення, вказаного у тайматуті переходу t4. За замовченням стоїть значення 7. Тобто протягом 7 секунд буде блимати зелений колір.
3. Наступний стан — зміна кольору сівтлофору. Загорається жовте світло, яке горить протягом таймауту у переході t5. За замовченням значення 4 говорить про те, що жовтий колір буде горіти 4 секунди. Збільшивши значення отримаємо довший попереджувальний сигнал для водіїв.
4. Далі загорається червоне світло для водіїв та пішоходам дозволено переходити дорогу. Тривалість зеленого кольору для пішоходів вказана у таймауті переходу t8 з діаграми пішоходи. За замовченням встановлено значення 25, тобто у пішоходів є 25 секунд, щоб перейти дорого. В разі, якщо цього часу недостатньо, його треба збільшити.
5. Коли час переходу закінчився, починає блимати зелений кольір для пішоходів, що попереджує ії про те, що скоро рух буде заборонено. Процес попередження проходить аналогічно з 2 експериментом.

Після цього для пішоходів загорається червоний колір, а для машин — зелений. Таким чином система пройшла через усі можливі стани та повернулася у початковий. Процес починається знову.

Індивідуальне завдання.

1 Змінити модель таким чином, щоб генератор створював «тики» у випадкові моменти часу.

Для цього пропишемо код, який представлено на рисунку 11.



Рисунок 11 – Код, який дозволяє створювати тики у випадкові моменти часу

Тоді працююча модель буде виглядати як на рисунку 12.

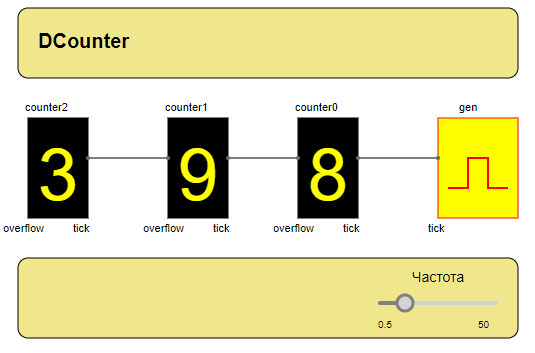


Рисунок 12 – Готова модель десяткового лічильника

2 Змінити модель таким чином, щоб частота миготіння зеленого світла регулювалася за допомогою слайдера.

Для цього створено змінну frequency, та зроблено деякі зміни в моделі як показано на рисунку 13.

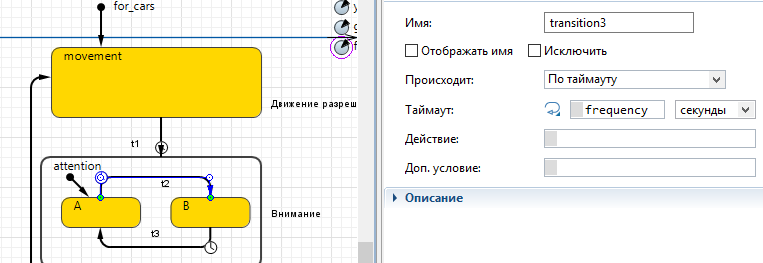


Рисунок 13 – Встановлено частоту миготіння як зміну

У загальному вигляді це буде виглядати, як показано на рисунку 14.

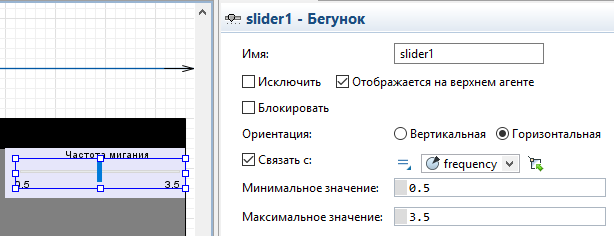


Рисунок 14 – Слайдер, що регулює частоту миготіння зеленого світла

3 Змінити модель автоматичного світлофора автомобілів і пішоходів (без кнопки), видаливши переходи по сигналу і підберіть відповідні таймаути для узгодженої роботи світлофорів.

Модель, що була розроблена раніше була перероблена як показано на рисунку 15-16.



Рисунок 15 — Діаграма станів для автомобілів

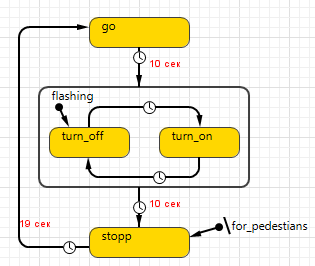


Рисунок 16 — Діаграма станів для пішоходів

У загальному вигляді модель буде виглядати, як на рисунку 17.

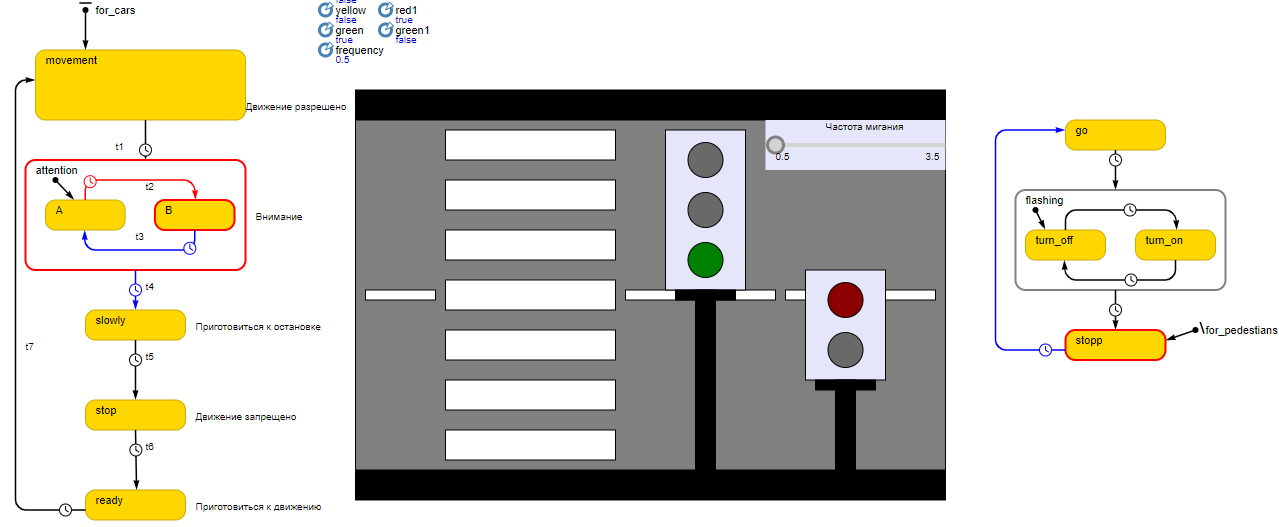


Рисунок 17 – Модель автоматичного світлофора автомобілів і пішоходів

**Висновки**

В ході виконання лабораторної роботи були отримані навички створювання нових класів активних об’єктів, роботи з портами, подіями та повідомленнями, розробки діаграм станів, в результаті моделювання десяткового лічильника, пішохідного та автомобільного переходів.