

Національний технічний університет України «КПІ ім. Ігоря Сікорського»  
Факультет Інформатики та Обчислювальної Техніки  
Кафедра інформаційних систем та технологій

Лабораторна робота №5  
з дисципліни «Технології інтернету речей»

на тему

«Реалізація IoT-проекту»

Виконала:  
студент групи ІП-11  
Дякунчак І.

Викладач:  
В. А. Нікітін

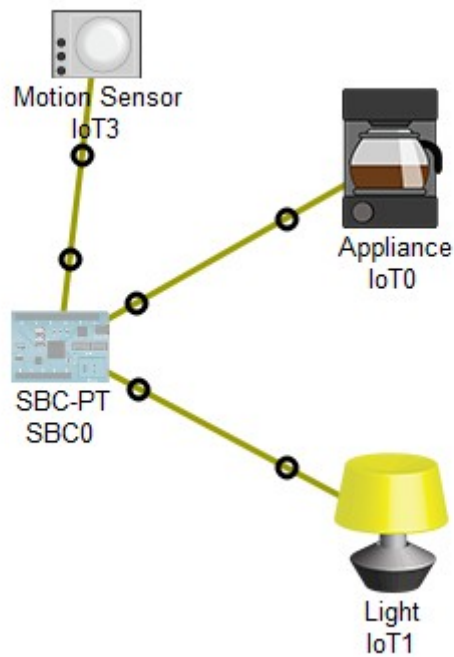
Київ – 2024

**Мета роботи** — ознайомитись з візуальним програмуванням мікроконтролера та реалізувати IoT-проект з використанням Cisco Packet Tracer .

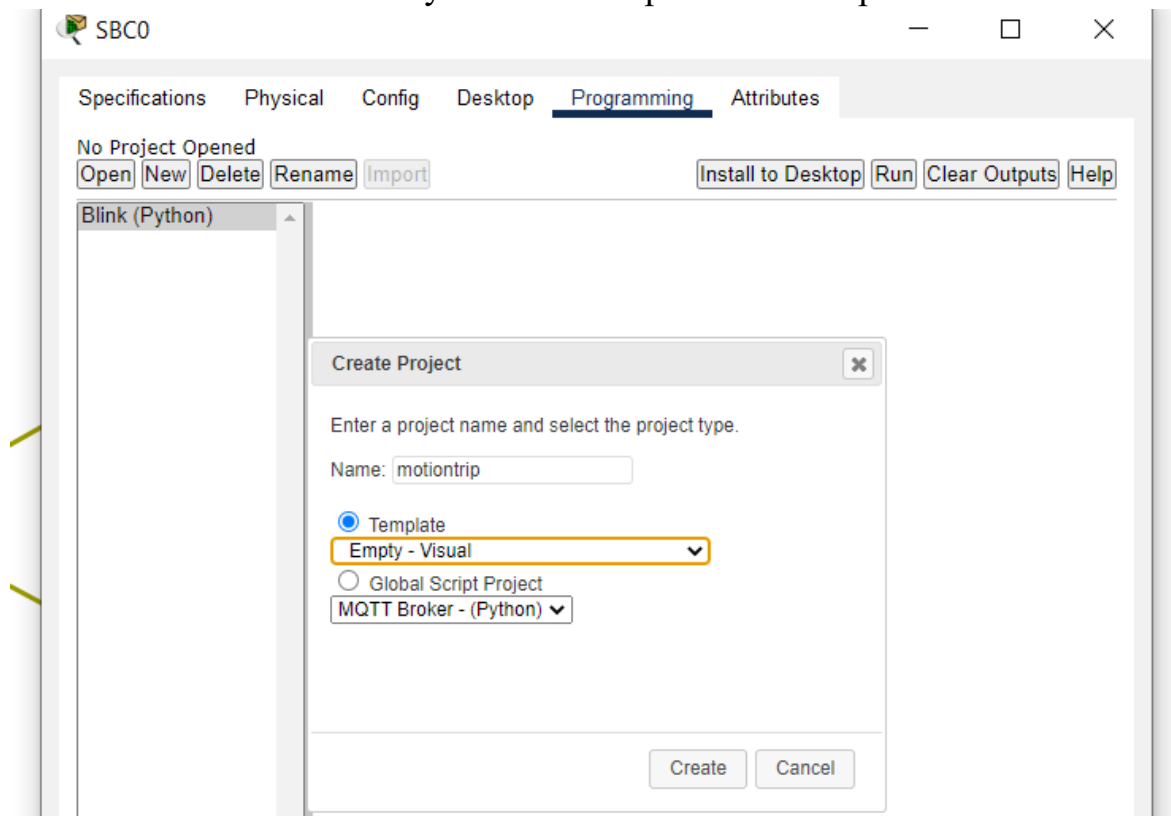
## **Виконання:**

**Завдання 6.1.** Підключення та візуальне програмування мікроконтролера

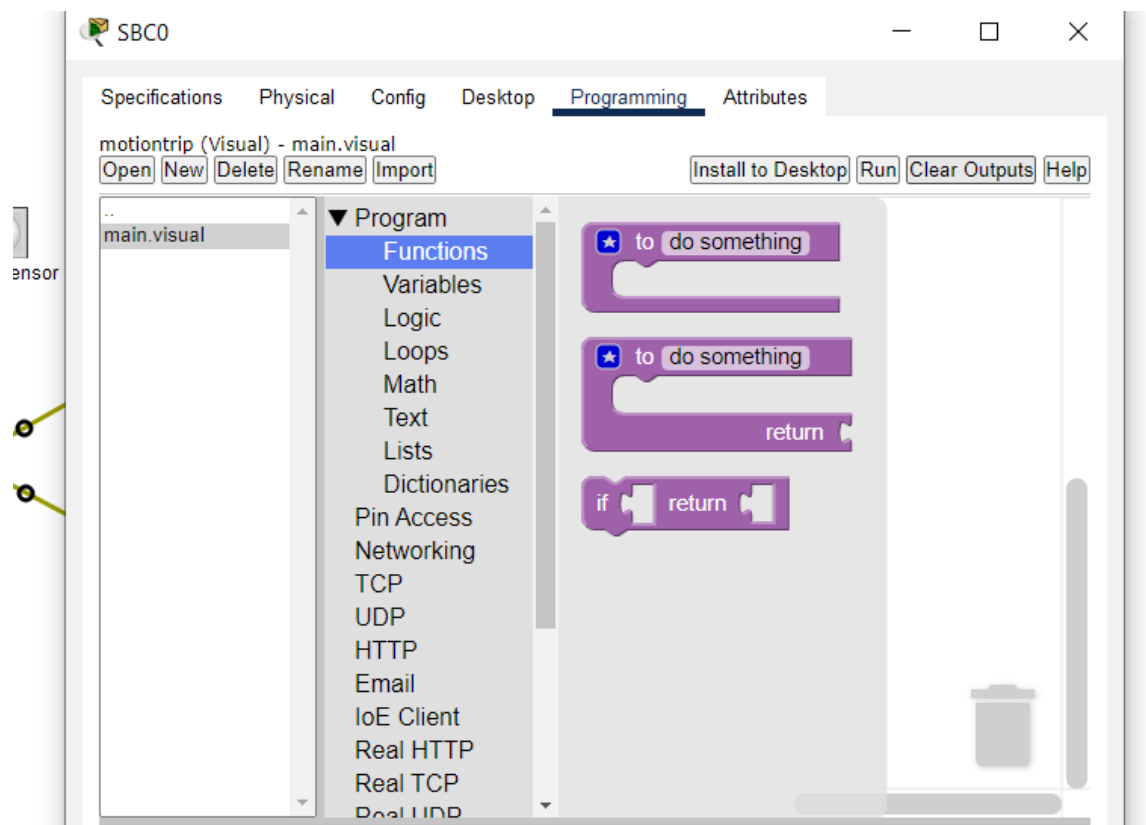
Складемо схему із SBC мікроконтроллером.



Запрограмуємо SBC. Для цього, відкриваємо властивості Програмування на SBC0. Клікнемо на «Blink Python» та створимо новий проект.

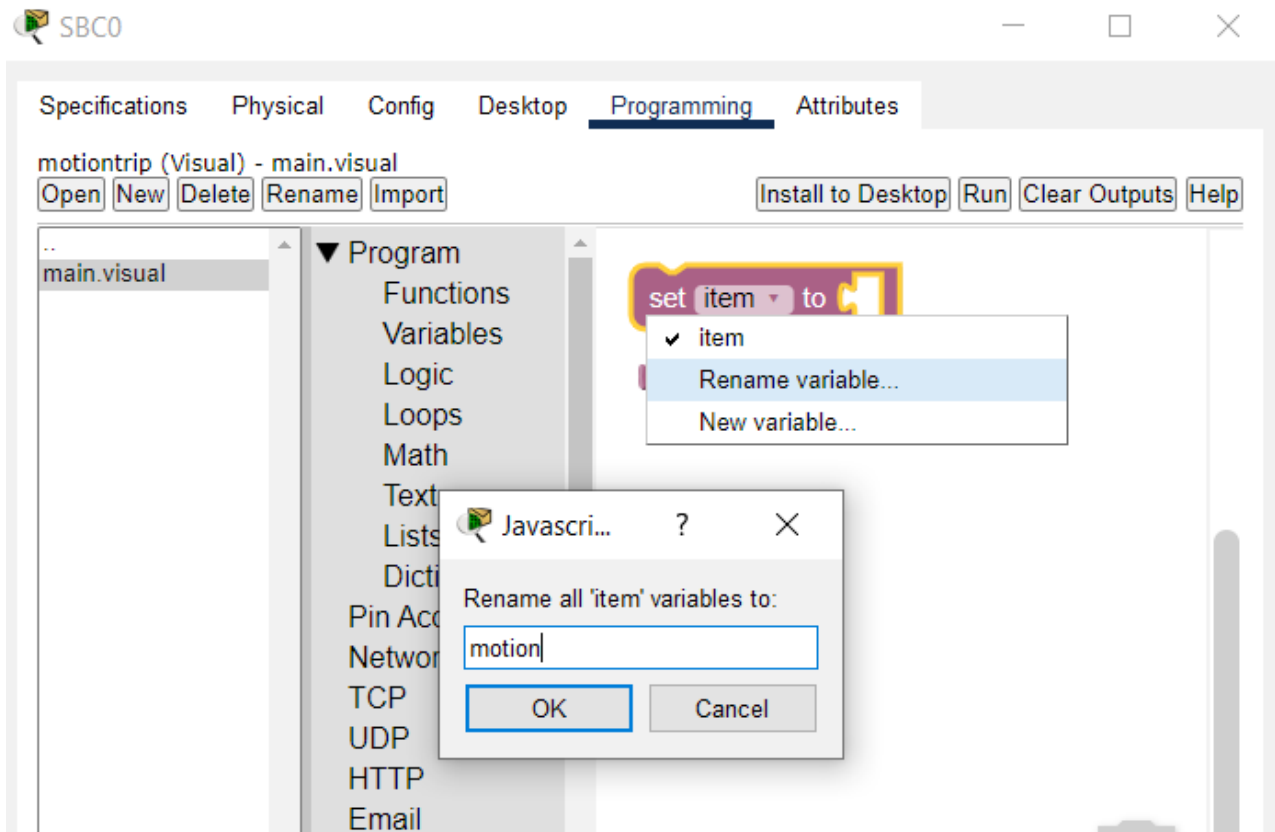


Зробимо подвійний клік на «main.visual», а потім Program та ознайомимось з елементами візуального програмування у цьому середовищі.

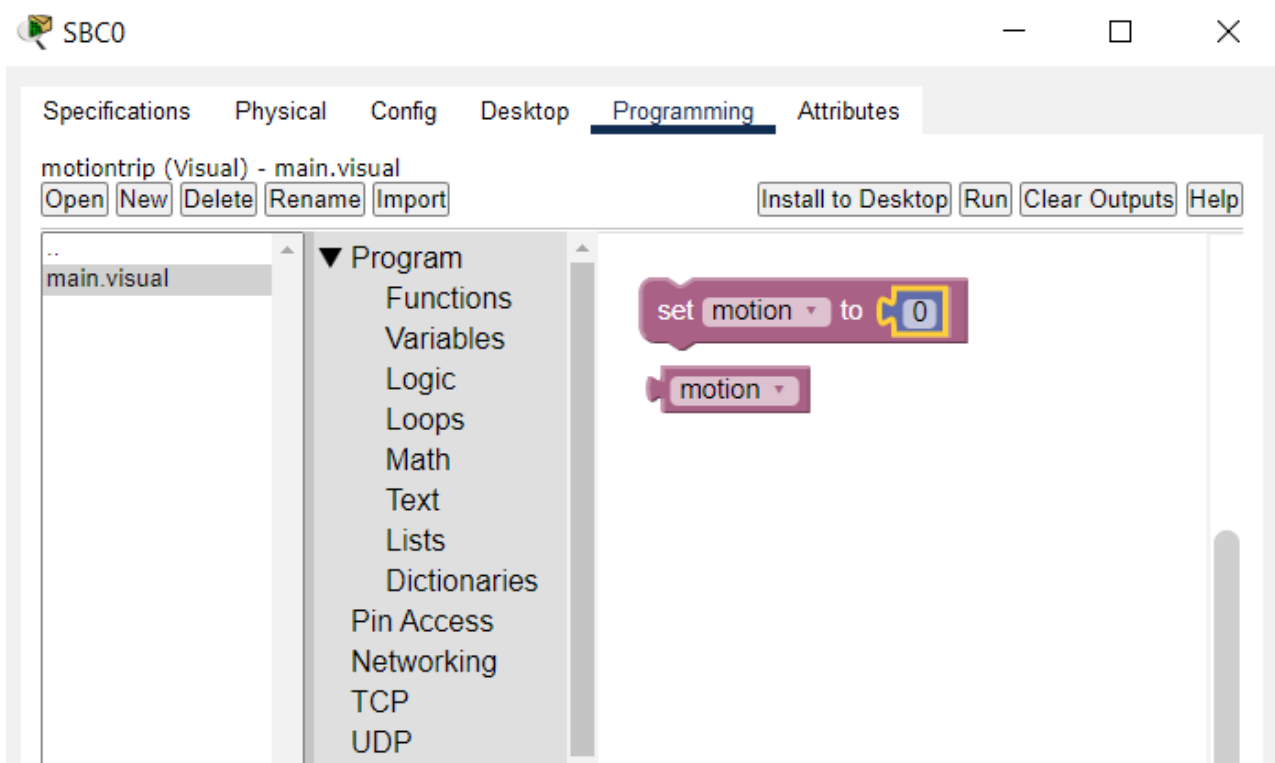


Обираємо «Змінні» - Variables та перетягуємо елемент на робочу область.

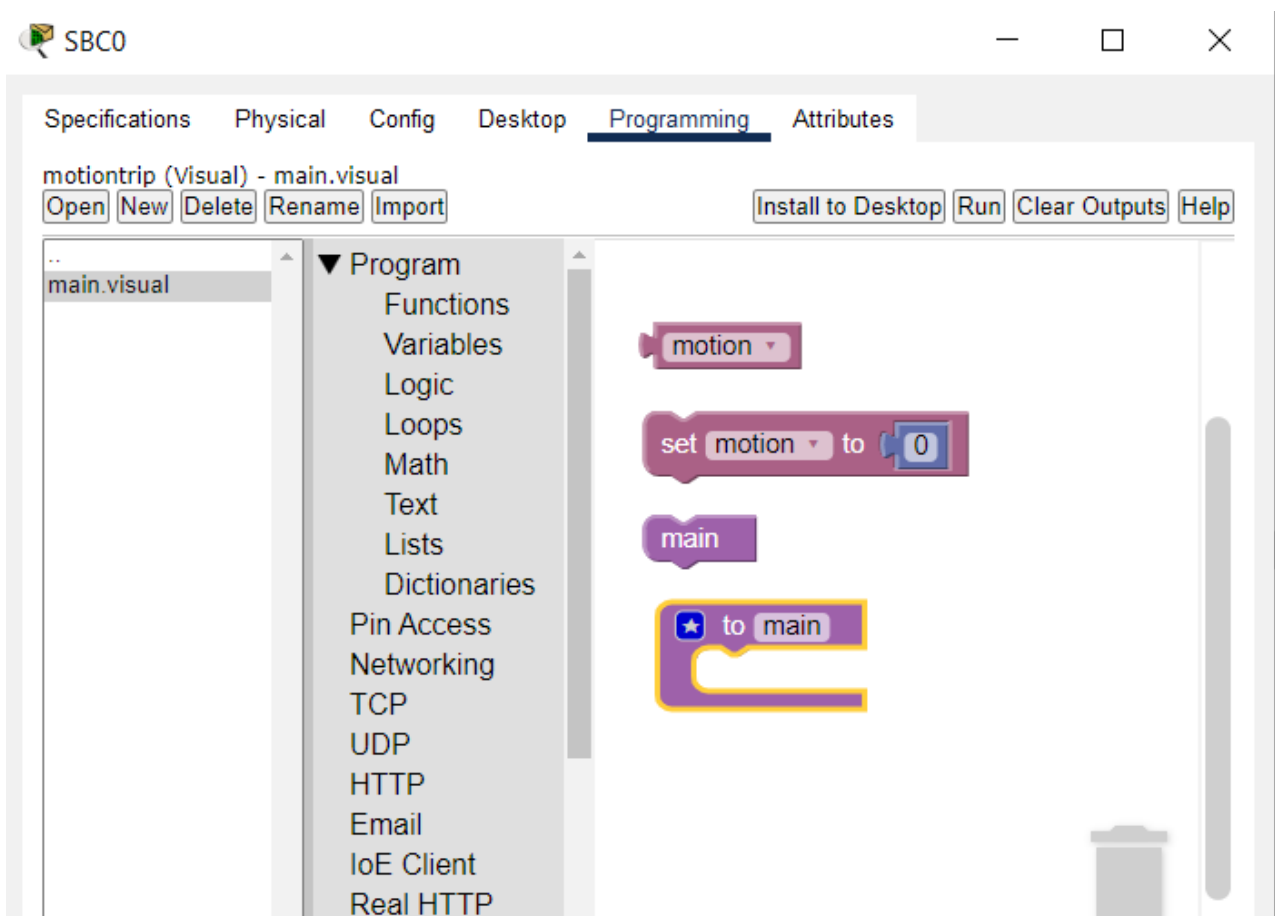
Змінимо назву змінної.



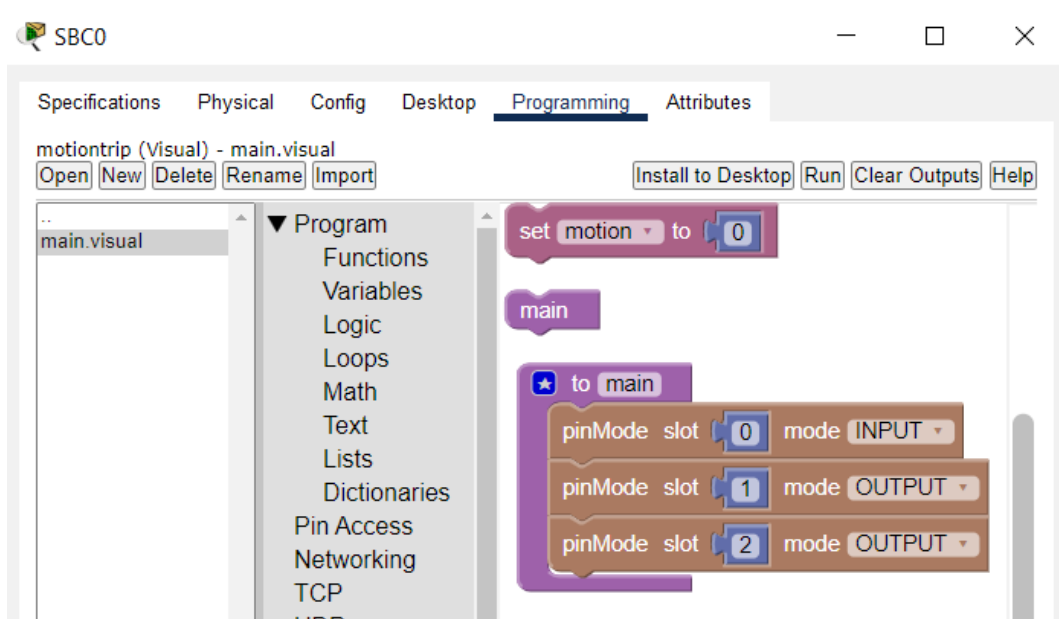
Далі розташуємо елемент “0” з підрозділу “Math” і вставимо його в чарунку.



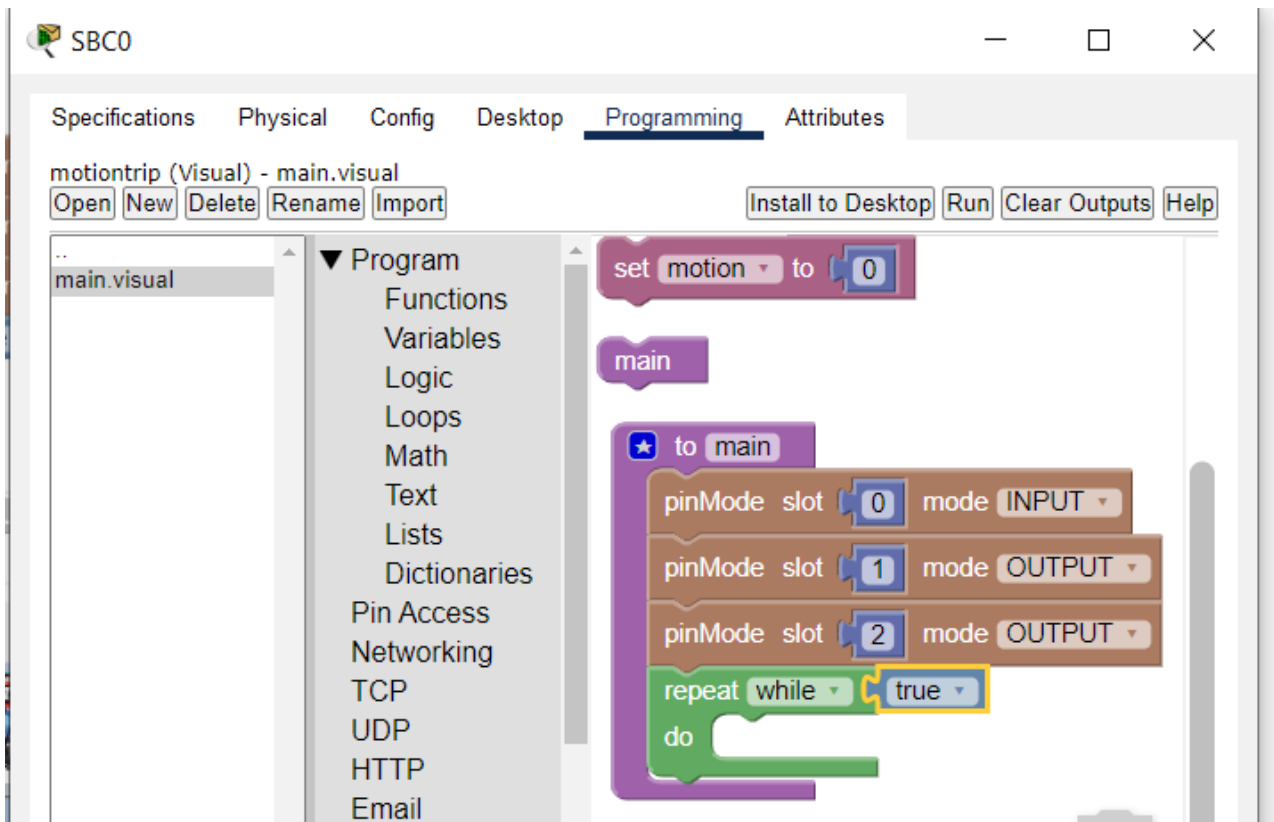
З підрозділу функцій розташовуємо і встановлюємо початок основної функції “Main”. Змінюємо назву з “do something” на “main”.



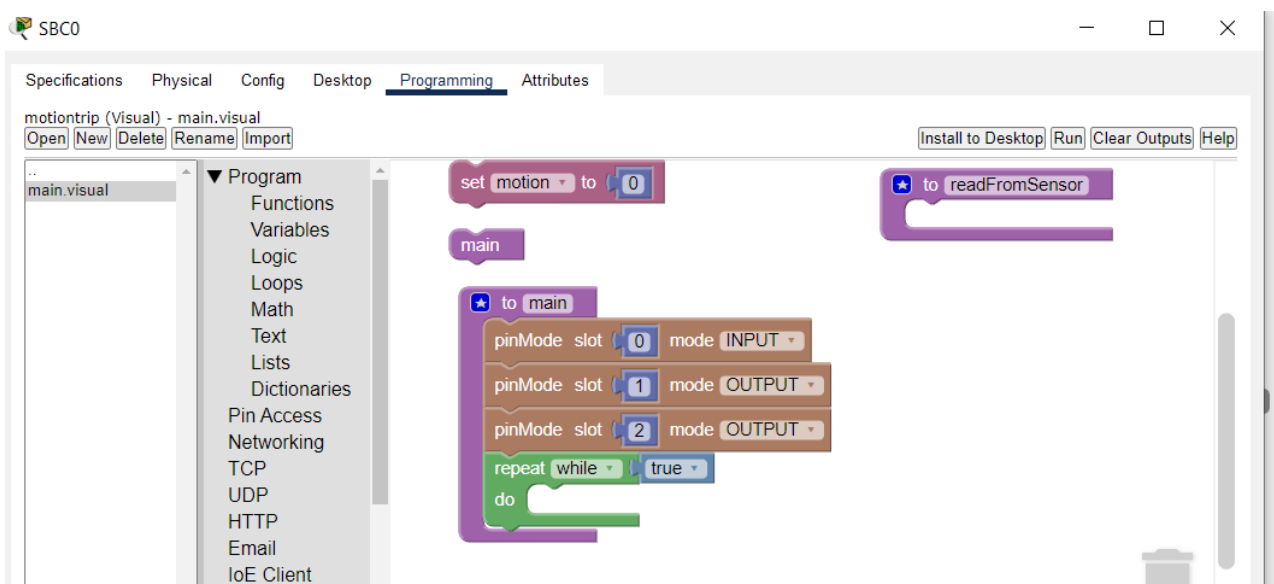
З Pin Access дістаємо функцію pinMode для розподілу контактів. Створюємо копії цього екземпляру (всього 3 штуки), налаштовуємо входи/виходи, і додаємо піни в тіло функції.



Додаємо функцію циклу з підрозділу Loops та функцію true з підрозділу Logic. Включаємо їх до функції main.

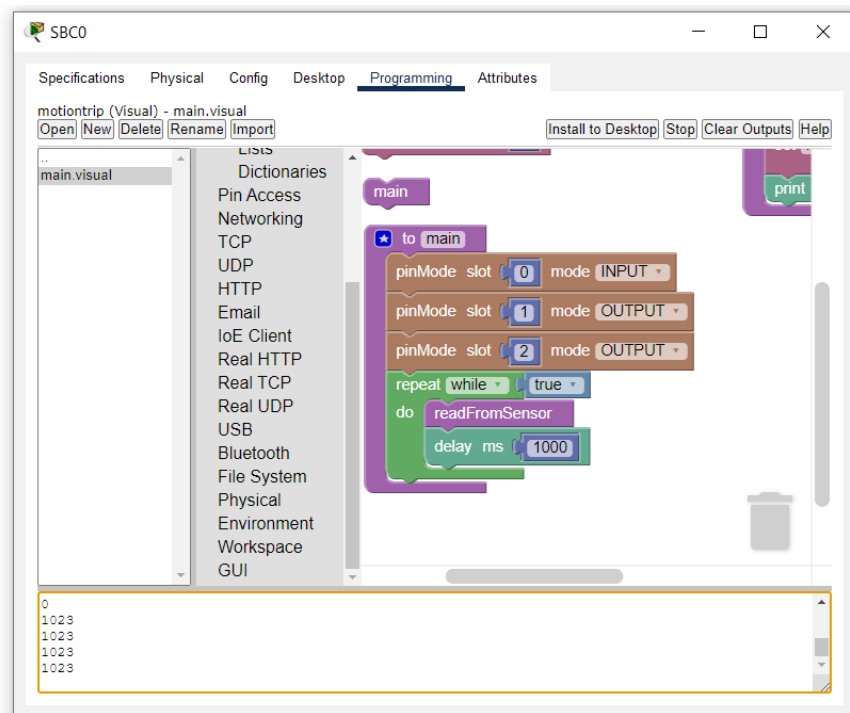
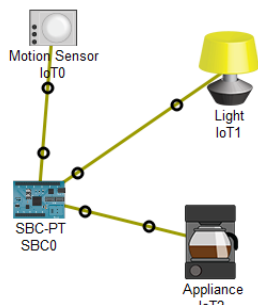


Розташуємо функцію з розділу Functions, яка буде зчитувати сигнали з сенсорів. Змінюємо її назву.

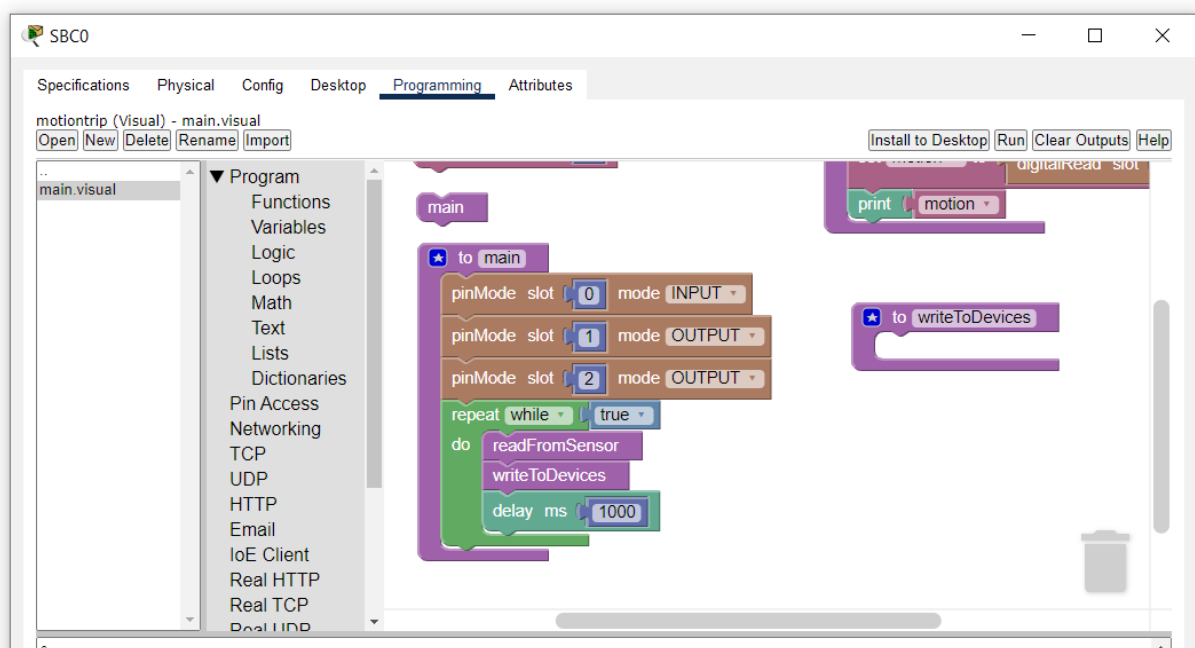


Physical x 2219, y: 288





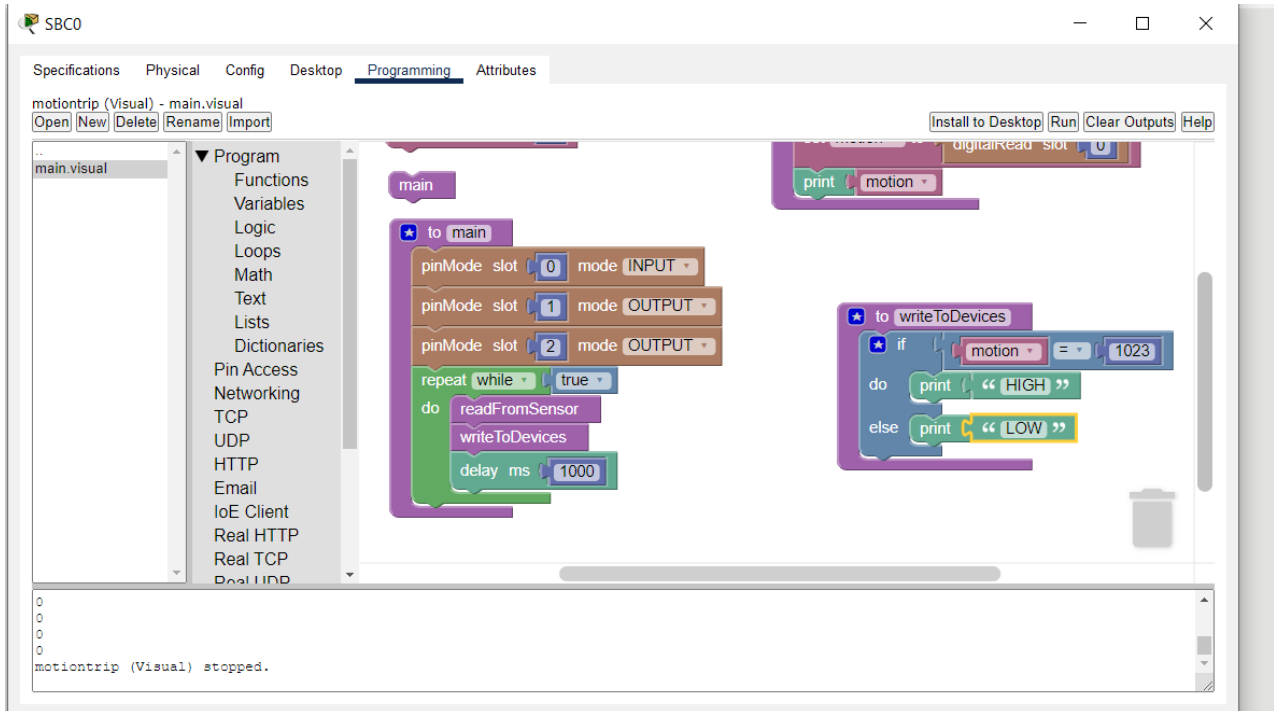
Створимо підпрограму для управління пристроями IoT. Формуємо функцію запису. Додаємо її до основної програми.



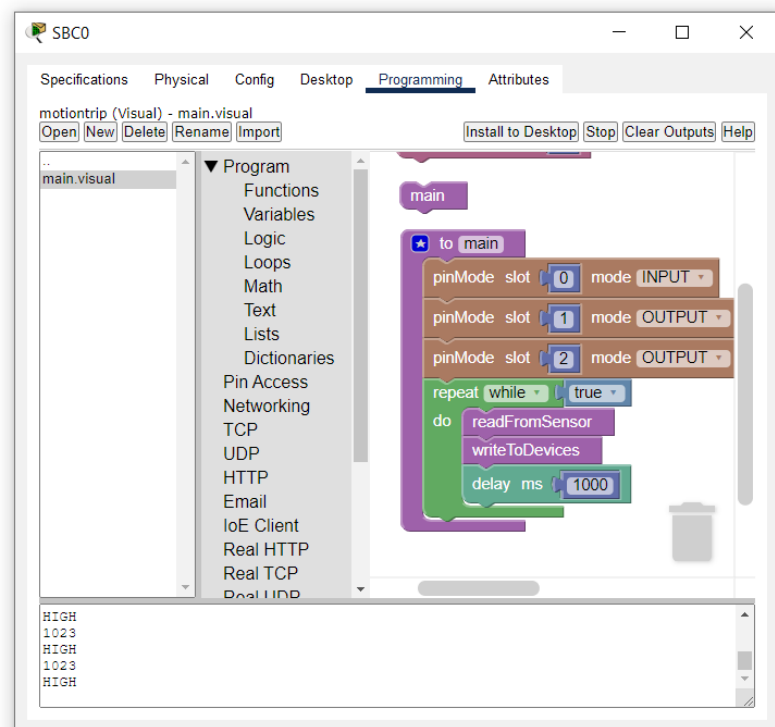
Далі необхідно забезпечити виконання умови: після читання сигналу з датчику (D0), потрібно запустити виконання певної дії. Нехай такою дією буде запис активного рівня в D1 та D2. Додаємо в робочу область функцію if-do-else та розташовуємо її в підпрограму запису. Перетягуємо функцію умови з підрозділу



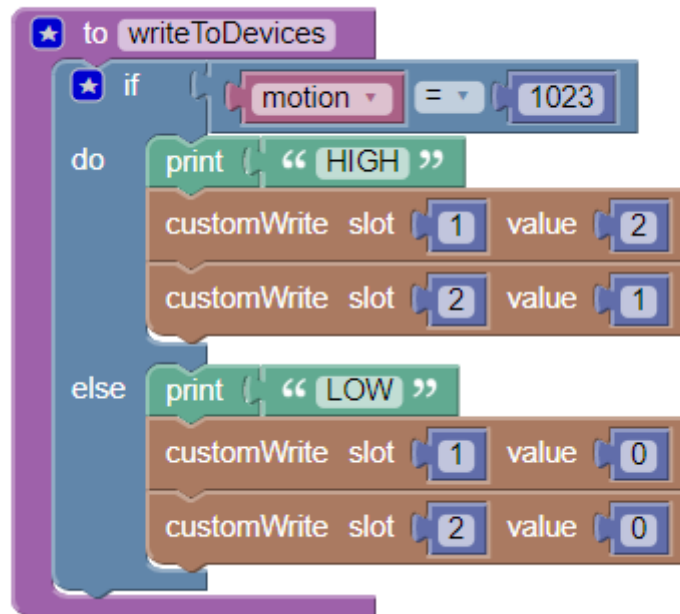
Logic та додаємо до тіла функції. Розташовуємо елементи «print» та «лапки» з підрозділу Text. Прописуємо відповідні повідомлення.



Перевіримо роботу підпрограми. Для цього натиснемо кнопку Run та перевіряємо роботу детектора як в минулий раз.

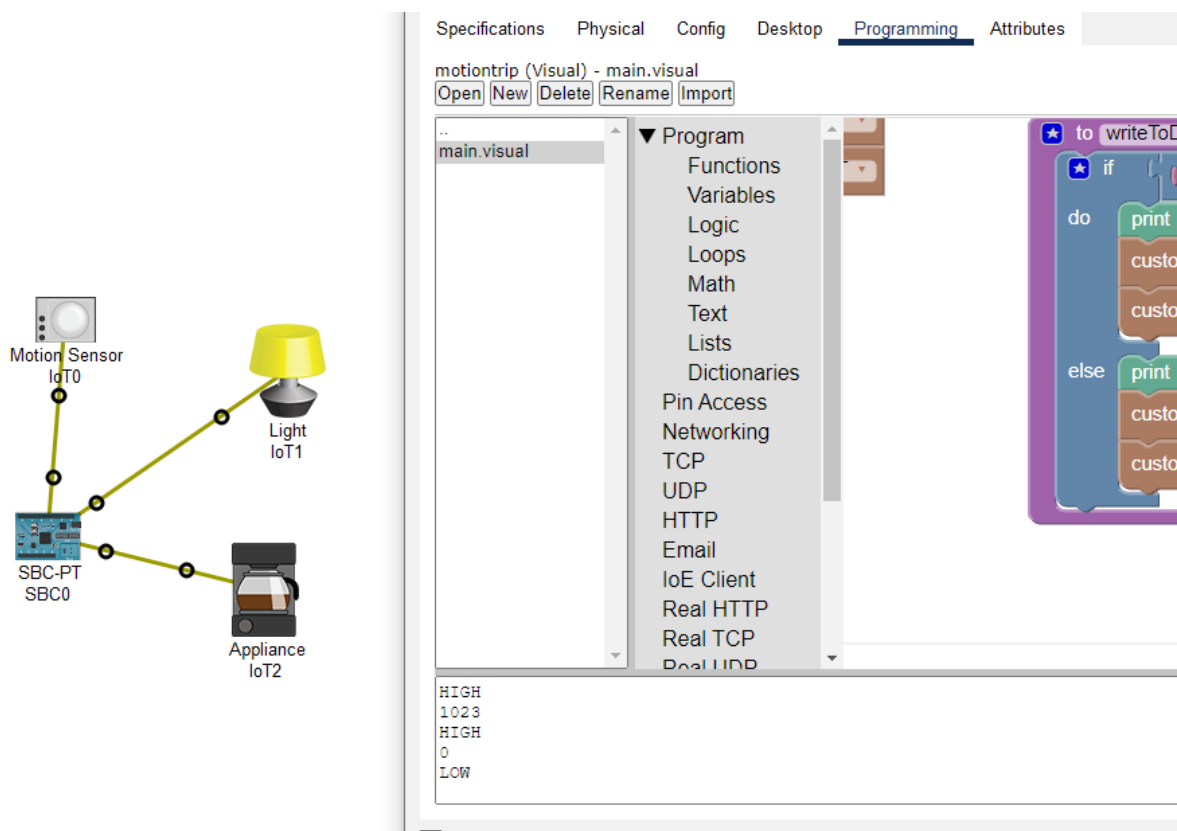


Реалізуємо функцію керування пристроєм. Дістаємо функцію CustomWrite з розділу Pin Access.



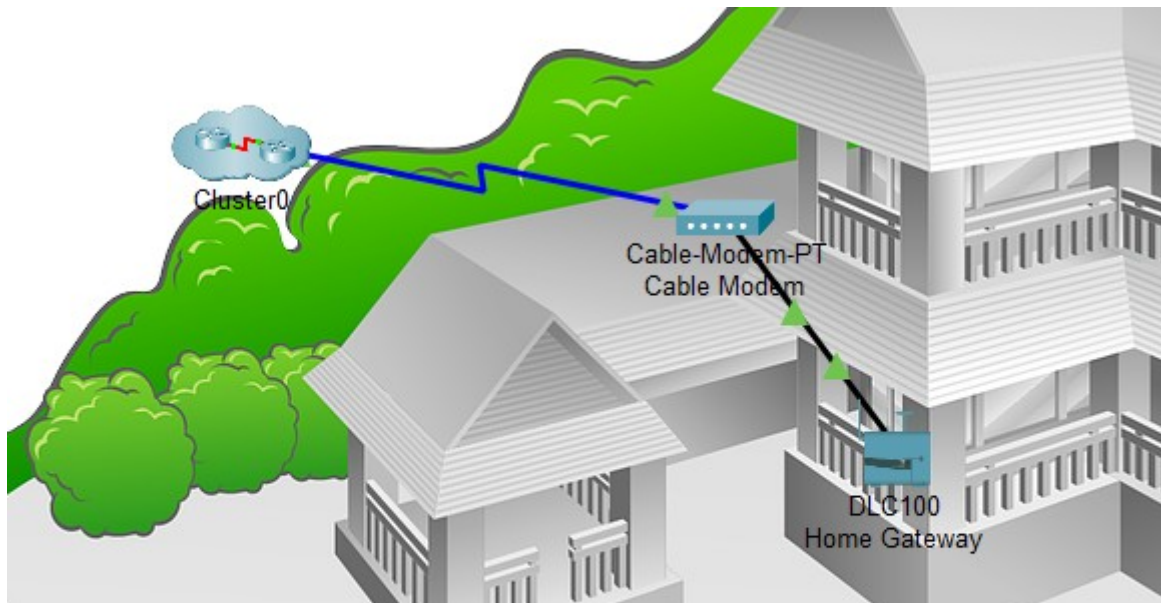
Запускаємо програму на виконання. При

спрацюванні датчика руху повинні вмикатися лампа та кавоварка. Після роботи вони вимикаються.

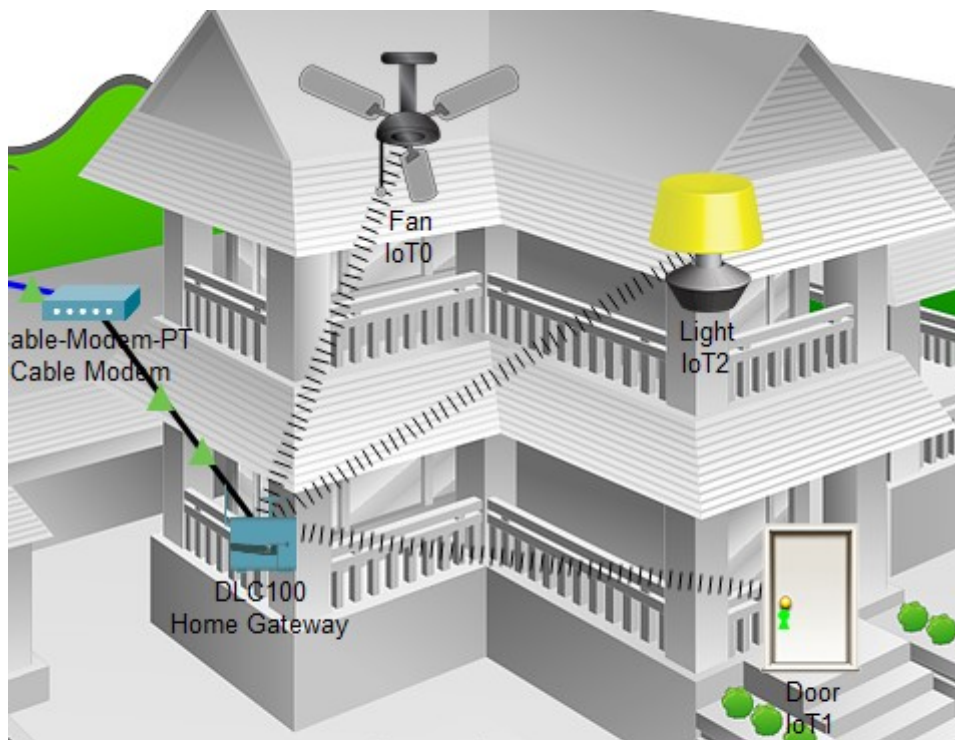


**Завдання 6.2.** Реалізація IoT-проекту з програмуванням та підключенням домашнього шлюзу до мережі

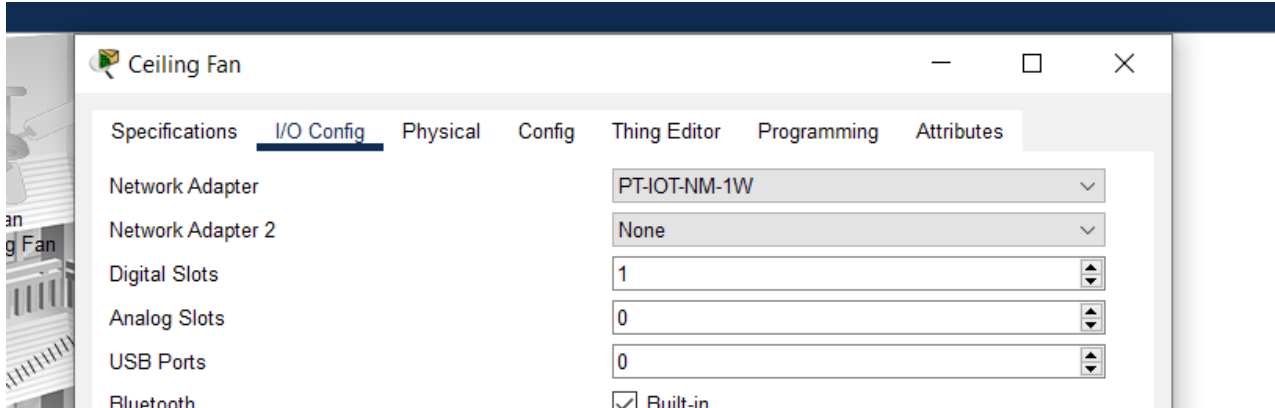
Додамо Home Gateway на робочу область. З'єднаємо Home Gateway з кабельним модемом, а модем з портом Інтернету. Через декілька секунд на кінцях ліній мають бути зелені позначки. Це підтвердження того, що ланки працюють.



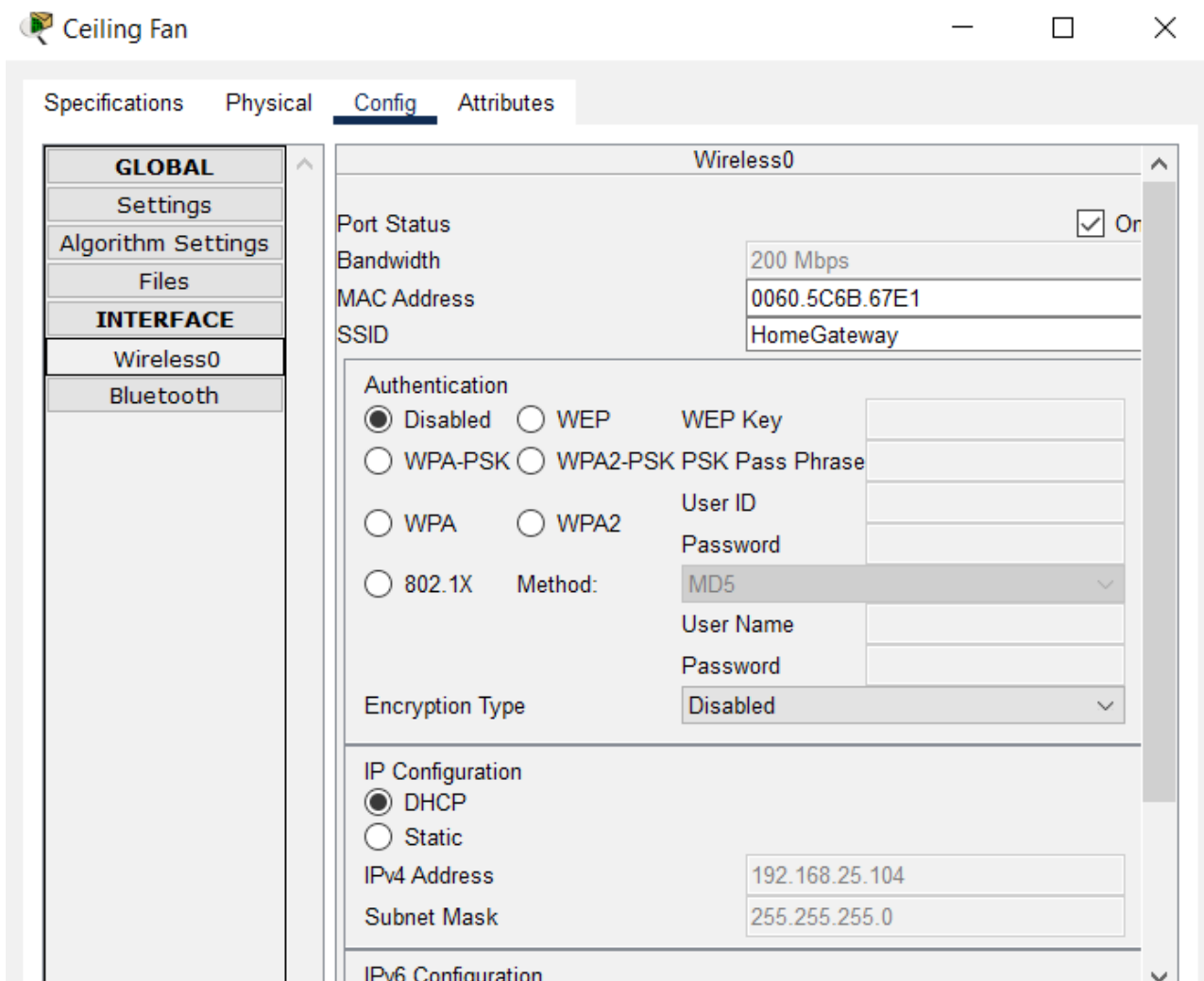
Додамо "Вентилятор", "Двері" та "Лампа" до робочого простору.



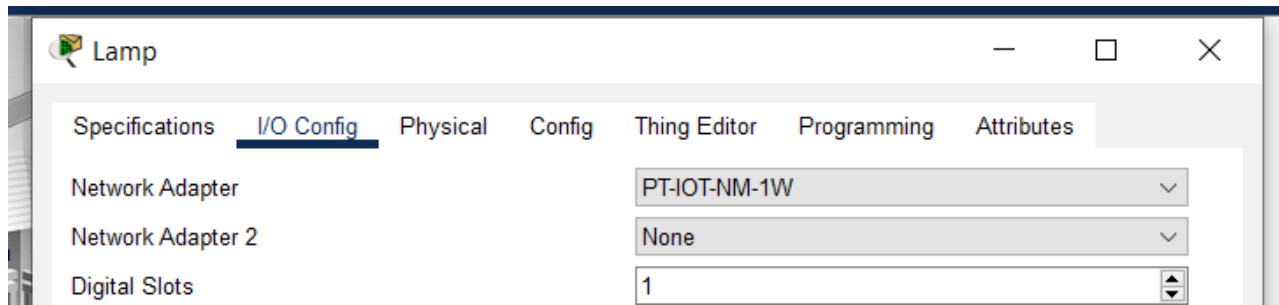
Налаштуємо бездротовий адаптер на вентиляторі та встановимо назву для Вентилятора Ceiling Fan.



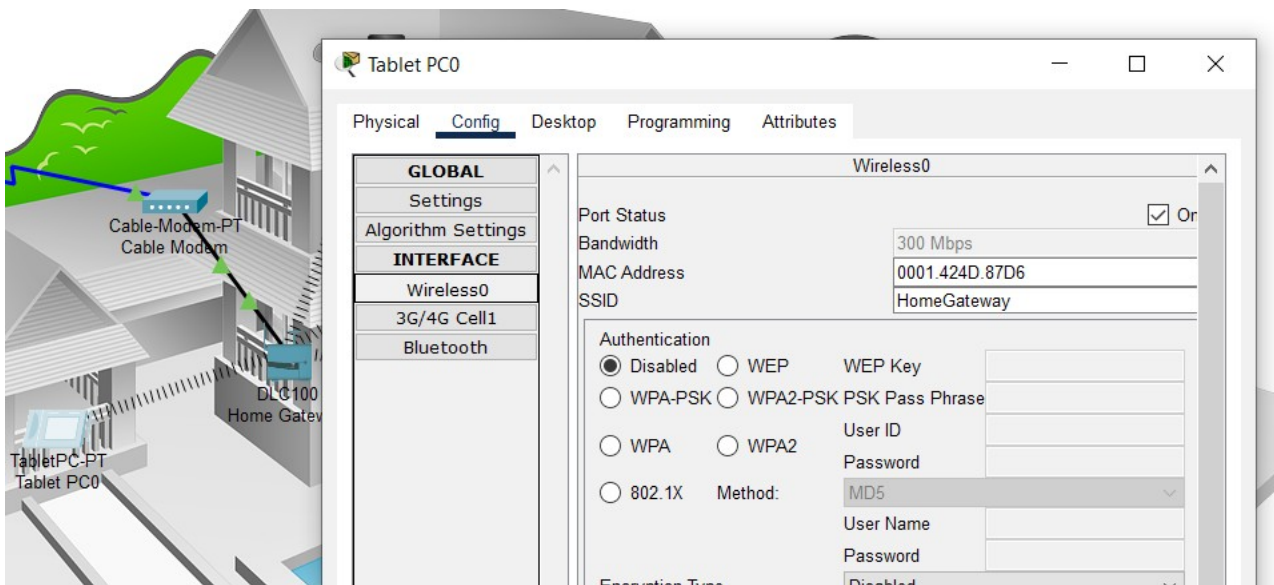
Перевіримо чи обрано DHCP в налаштуваннях IP-адреси, IP-адреса - 192.168.25.104 та Шлюз за замовчуванням - 192.168.25.1. Це вказує на те, що вентилятор підключено до мережі і є отримання IP конфігурації з домашнього шлюзу.



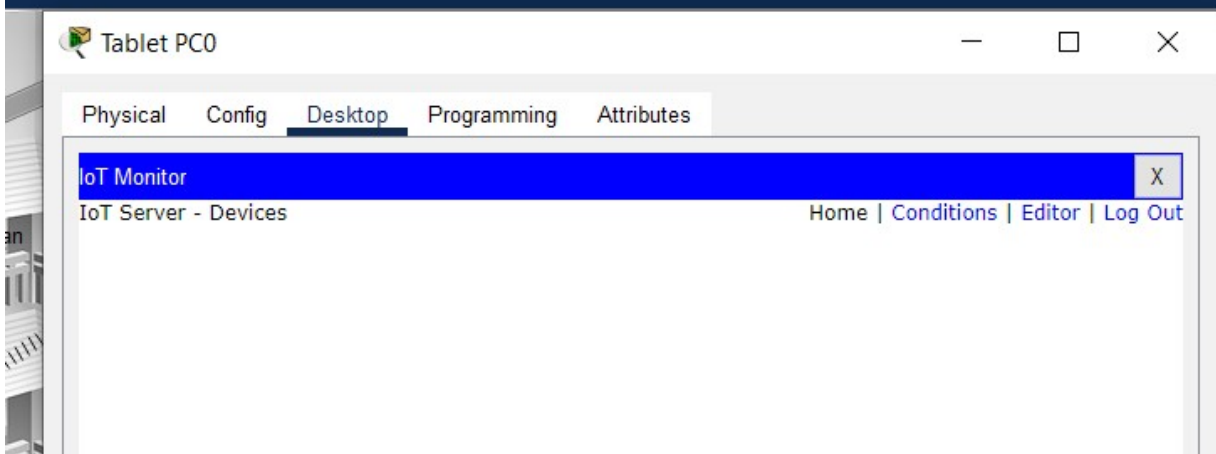
Так само підключимо двері та лампу до бездротової мережі.



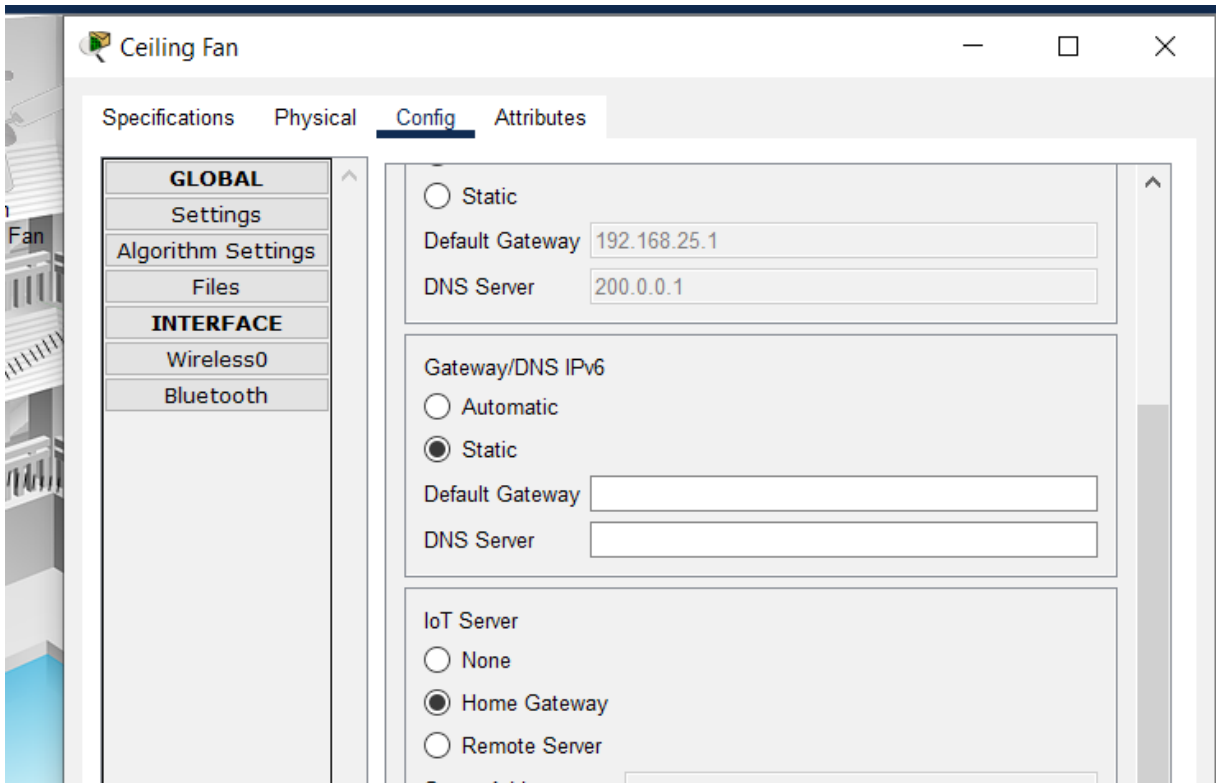
Додаємо і підключаємо до мережі планшет.



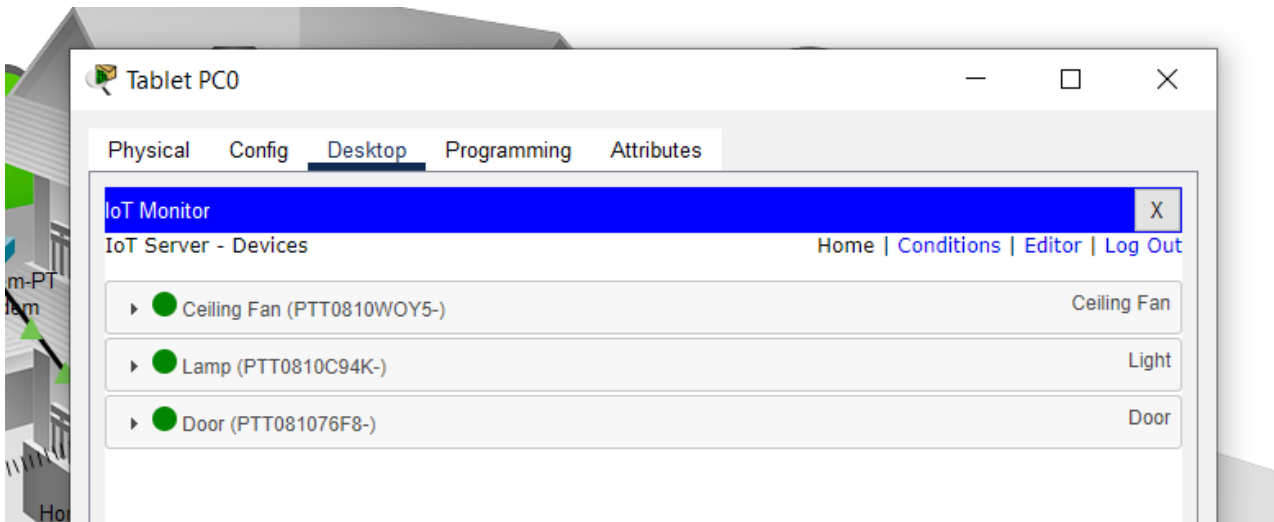
Переглянемо список підключених пристроїв.



Як бачимо, в списку не відображається жоден IoT-пристрій, тому налаштуємо всі пристрої на те, щоб вони підключились до IoT Server.

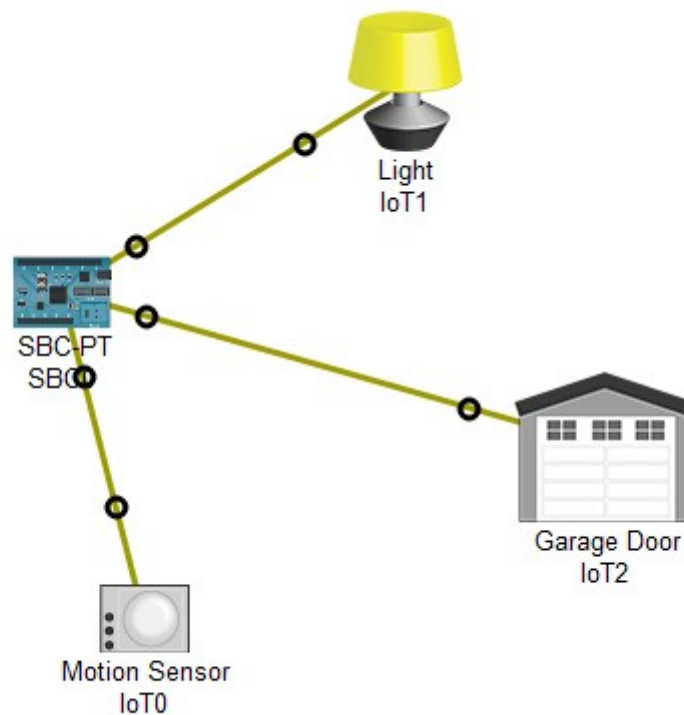


Перевіряємо чи тепер відображаються підключені пристрої.



### Завдання 6.3. Реалізація IoT-проекту з використанням SBC

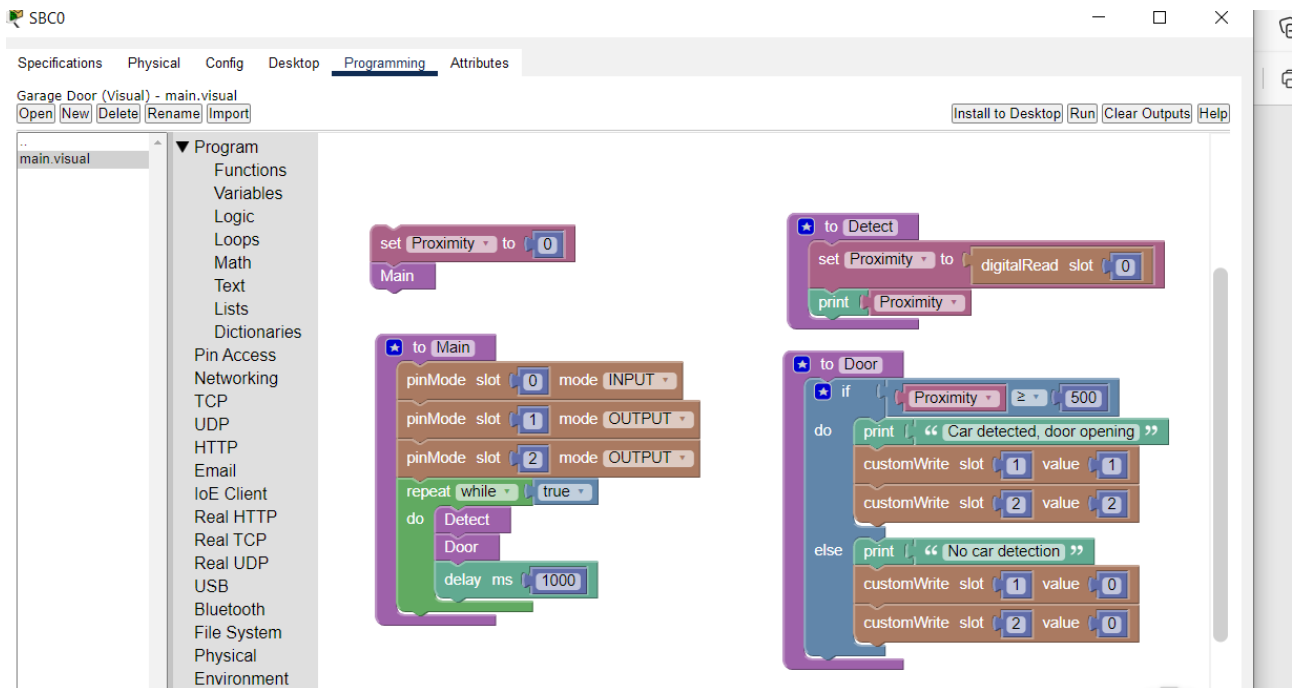
Реалізуємо наступну схему. Всі компоненти підключаємо через спеціальні кабелі IoT до SBC пристрою.



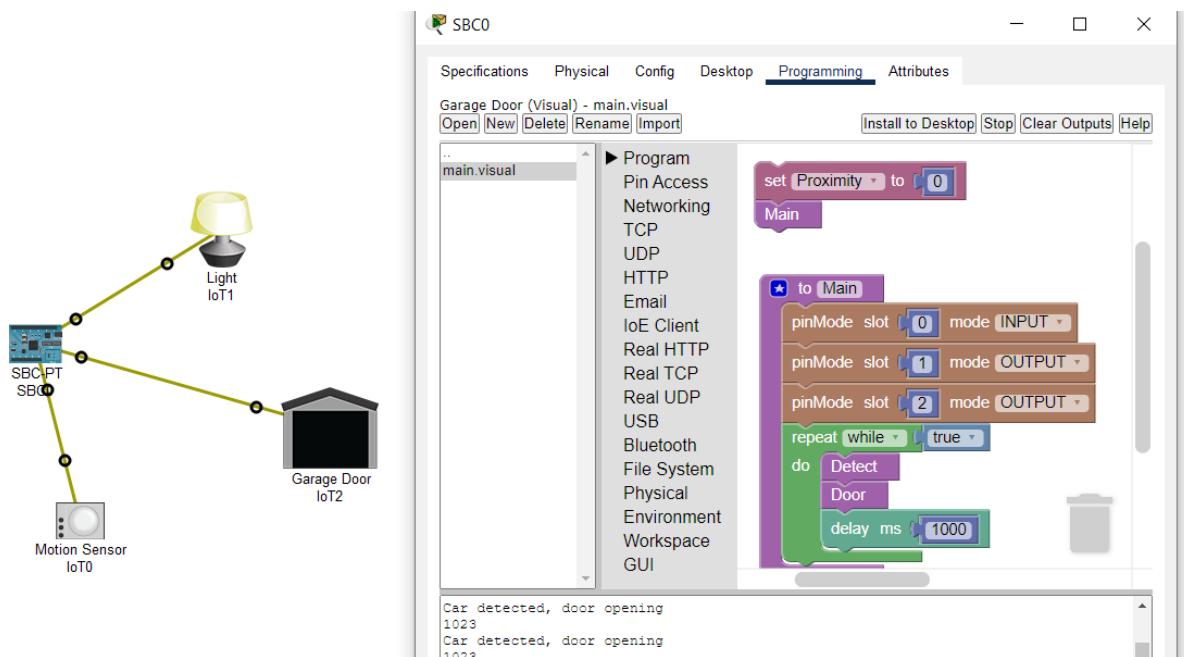
Приклад імітує відкриття гаражних дверей разом із включенням світла, коли Датчик руху виявив певний рух. У цьому випадку логіка симуляції не була



реалізована сервером IoT, а індивідуальним програмним забезпеченням, що працює на платі SBC. Логіка програмного забезпечення дуже проста, сенсор був безпосередньо підключений до входу SBC. Коли об'єкти були виявлені, датчик подавав величину вхідного штифта SBC, Контролер потім порівнював його із заздалегідь встановленим пороговим значенням, і якщо логічна пропозиція була мет, команду до вихідних штифтів було відправлено. Команда була користувацькою командою API що відкрив гаражні двері та включив вогні.



Перевіримо чи коректно працює створена програма.





## Контрольні запитання

1) Які переваги та недоліки візуального програмування?

*Переваги:*

- Легкість використання
- Швидше прототипування
- Збільшення продуктивності

*Недоліки:*

- Обмеження в гнучкості
- Залежність від інструментів
- Можливість переплутування логіки

2) Чи підтримує Cisco Packet Tracer якісь мови програмування?

Cisco Packet Tracer не є середовищем для повноцінного програмування, однак підтримує деякі базові концепції програмування, такі як скрипти на Python.

3) Для моделювання якого IoT-проекту ви використовували б Cisco Packet Tracer?

Для створення повністю автоматизованого будинку, в якому пристрої, такі як освітлення, вентиляція, опалення, системи безпеки, і побутова техніка автоматично реагують на умови навколишнього середовища або команди користувача; для моделювання міської інфраструктури з інтелектуальними системами управління вуличним освітленням, сміттєвими баками, парковками та системами моніторингу руху.