Умный дом 1 был первым из двух симуляций, которые охватили домотическую область IoT. Фактически интеллектуальные устройства были подключены к IoT, чтобы имитировать полное взаимодействие компонентов и возможность удаленного управления устройствами. Владелец дома фактически, после подключения через браузер и прохождения аутентификации, смог командовать гаражом или регулировать вентиляцию дома, а также проверять текущее состояние охранной системы или уровень углекислого газа в гараже.

Этот случай также предоставил студентам возможность расширить симуляцию, используя сеть удаленного корпоративного офиса, для создания удаленного доступа к домашней локальной сети.

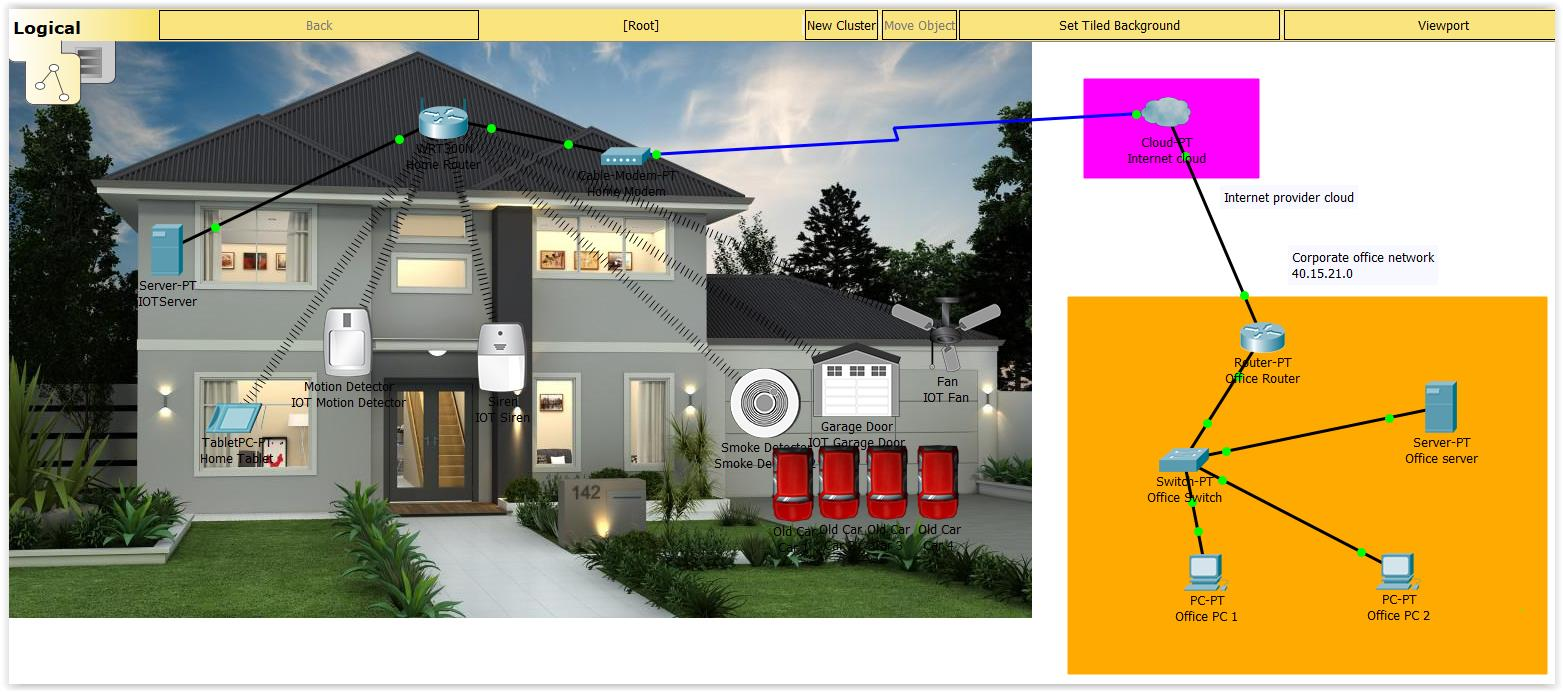


Рисунок 16 - Схема Cisco Packet Tracer для модели Smart-Home 1

Расположение сети

Это упражнение имело самую простую настройку по сравнению с другими симуляциями.

Сеть была логически разделена на три области: домашняя сеть, ISP Cloud и корпоративная офисная сеть.

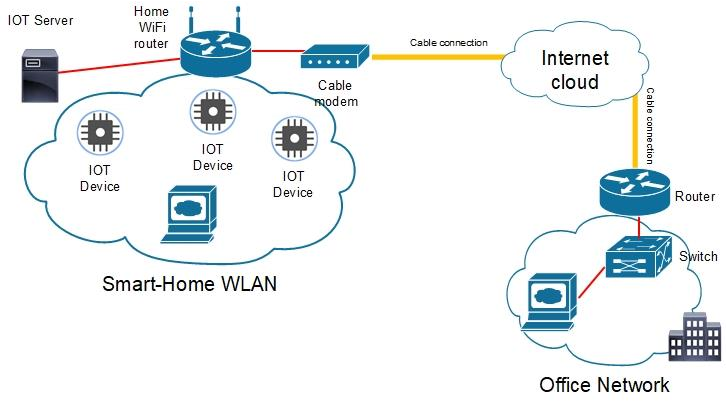


Рисунок 17 - Топология сети Smart-Home 1

Как видно из рисунка 17 выше, в центре этой симуляции была домашняя сеть, все IoT-устройства, домашние планшеты и серверы IoT были подключены к домашней WLAN. Сеть имитировала обычную домашнюю беспроводную сеть, где каждое устройство было подключено к беспроводному маршрутизатору, затем маршрутизатор был подключен к выделенному кабельному модему. В этом упражнении был необходим модем, так как беспроводная маршрутизация была оснащена только портами Ethernet, а подключение к ISP обеспечивалось через коаксиальный кабель.

Интернет-настройки маршрутизатора были установлены на значения по умолчанию для Интернет-провайдера DHCP, однако внутренний WLAN DHCP был отключен через графический интерфейс пользователя (GUI), поскольку локальный сервер, подключенный к WLAN, функционировал как DHCP. Только другими настройками в маршрутизаторе WLAN были домашний беспроводной SSID и пароль.

Все беспроводные устройства должны были использовать те же SSID, пароль и настройки DHCP по умолчанию, за исключением локального сервера, который использовал статический IP-адрес 10-класса.

Статические IP-адреса гарантировали, что, даже если маршрутизатор WLAN был перезагружен, IP-адрес сервера оставался прежним, без необходимости перенастраивать устройства с новым IP-адресом IoT-сервера. Сервер, в дополнение к службам DHCP, как было упомянуто ранее, также предоставлял функции IoT и DNS. Функции IoT были необходимы для того, чтобы предоставить бэкэнд-интеллект для моделирования IoT и для возможности размещения домашней страницы IoT, где домашний пользователь мог подключиться. Служба DNS также требовалась для преобразования URL-адреса домашней страницы IoT в IP-адрес собственного IoT-сервера.

Вторая сеть, представленная в упражнении, была облаком ISP. Это была искусственная имитация сервера ретрансляции, который создавал связь между двумя отдельными интерфейсами, сервер был настроен на подключение коаксиального кабеля от домашней модемной сети к кабелю Ethernet, идущему от офисного маршрутизатора. Дополнительные параметры конфигурации в этом устройстве запрещены.

Корпоративная сеть офиса была базовой установкой с маршрутизатором, подключенным как к

Интернет-провайдер, а также локальный коммутатор, к которому подключены также два ПК и один сервер. Сетевые адаптеры маршрутизатора были установлены с IP-адресом ISP в одном интерфейсе и IP-адресом локальной сети в другом, базовый протокол RIP также использовался в настройке для обеспечения возможности соединения между двумя сетями.

Офисный коммутатор и офисные ПК использовали стандартную конфигурацию по умолчанию. Локальный сервер был настроен на использование статических IP-адресов и обеспечение функций DHCP в локальной сети офиса.

План IoT

Для этого моделирования, как упоминалось ранее, все устройства IoT и сервер IoT были подключены к одной и той же сети WLAN.

Логическое соединение IoT было установлено на вершине сетевого подключения.

Как показано на рисунке 18, все устройства IoT должны быть настроены для удаленного подключения к серверу IoT с использованием ранее созданного имени пользователя, пароля и IP-адреса сервера.

Успешное подключение было установлено, когда кнопка «Подключиться» сменилась на «Обновить».

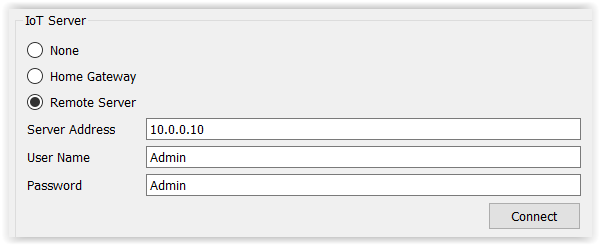


Рисунок 18 - Пример настройки IoT детектора движения

Все устройства должны использовать одинаковые учетные данные IoT, те же учетные данные также использовались владельцем дома для прохождения проверки подлинности при подключении через браузер к главной домашней странице мониторинга IoT, как показано на рисунке 19.

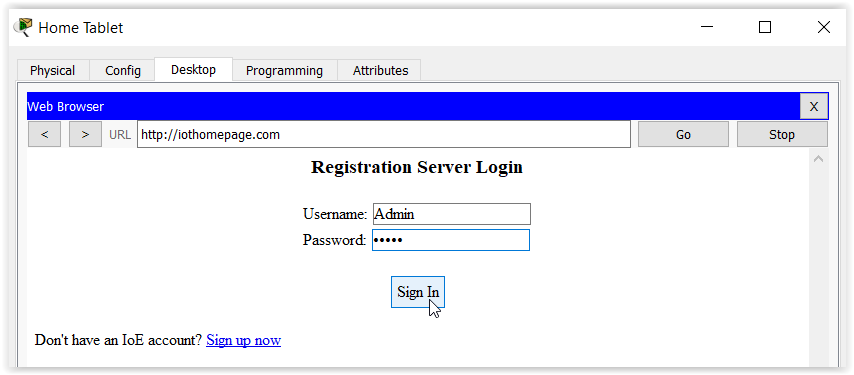


Рисунок 19 - Вход на домашнюю страницу IoT

Поскольку сервер IoT был настроен также для служб DNS, iothomepage.com был переведен со статического IP-адреса сервера IoT.

После подключения пользователя к IoThomepage.com стало возможным визуализировать состояние устройств IoT, но также можно просмотреть логику взаимодействия между ними.

Как видно на рисунке 20, в этом упражнении использовались пять интеллектуальных устройств IoT: детектор движения, сирена, дверь гаража, вентилятор и детектор дыма. Также использовались автомобили, но это было только для влияния на переменную среды моделирования.

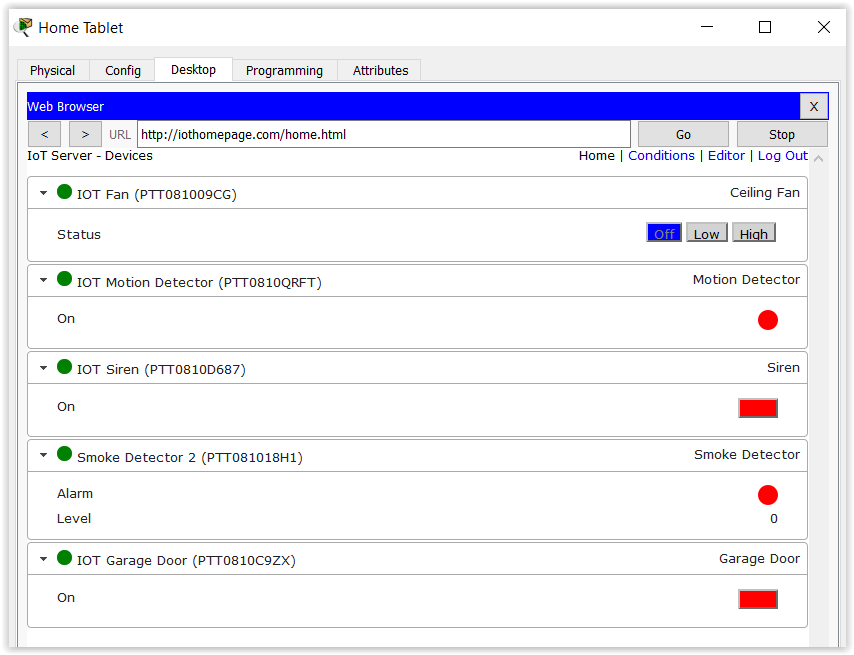


Рисунок 20. Состояние подключенных устройств IoT

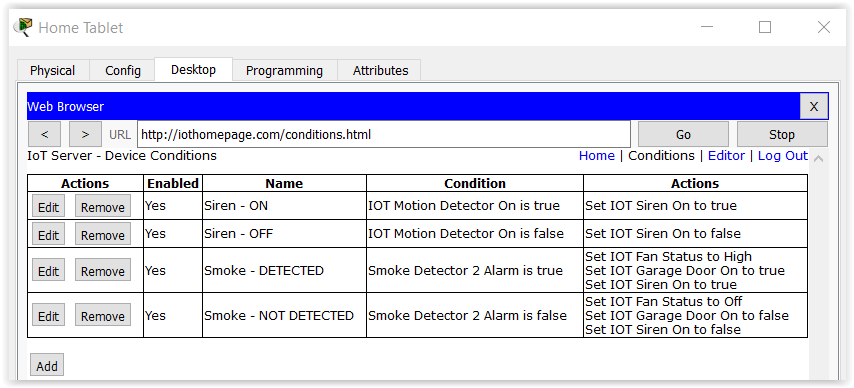


Рисунок 21 - Предварительно установленные условия

Как видно из рисунка 21, когда он был подключен к домашней странице IoT, он также был

можно перейти к странице условий и визуализировать два основных примера моделирования IoT.

В первом случае использовался датчик движения для временной активации сирены в качестве базовой системы домашней сигнализации. Логика видна на предыдущем рисунке 21, просто, когда датчик был включен, сирена была включена. Когда датчик не обнаруживал какого-либо движения и после предварительно установленного тайм-аута сирена была отключена. Эту логику легко проверить, нажав ALT на клавиатуре и наведя курсор мыши на детектор движения, затем сразу же активировали сирену, как показано на рисунке 22 ниже.

Пример IoT микроконтроллера

Один пример использования SBC также был включен в моделирование Cisco Packet Tracer. В этом сценарии ни один из датчиков или устройств не был подключен к домашней WLAN или какой-либо другой сети. Все компоненты были подключены через специальные кабели IOT к плате SBC.

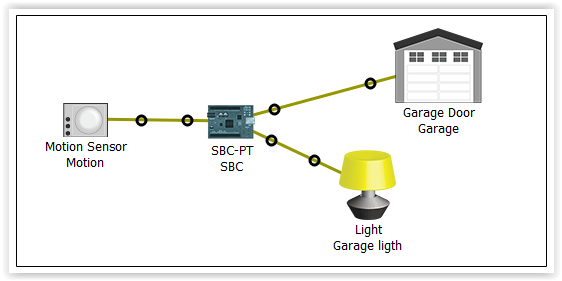


Рисунок 24 - Пример микроконтроллера Smart-Home 1

В примере имитировалось открытие ворот гаража и включение света, когда датчик движения обнаружил какое-то движение. В этом случае интеллектуальность моделирования обеспечивалась не внутренним IoT-сервером, а пользовательским программным обеспечением, работающим на плате SBC.

Логика программного обеспечения была очень простой, датчик был напрямую подключен к входному контакту SBC. Когда объекты были обнаружены, датчик подал значение на входной вывод SBC, затем микроконтроллер сравнил его с заданным пороговым значением, и, если логическое условие было выполнено, была отправлена команда на выходные выводы. Command была пользовательской командой API, которая открывала дверь гаража и включала свет.

Как ранее обсуждалось в главе 4.2, характеристики датчика, освещения и двери были видны на вкладке технических характеристик устройства.

Логика программы Blockly, запущенной на SBC, видна в Приложении 1.

Блочно настраиваемое программное обеспечение для IoT Simulation

