**Программное обеспечения для умного дома**

**{картинка}**

Всем добрый день. В качестве хобби написал программу, которую можно использовать как часть системы “умный дом”. Чтобы сразу заинтересовать читателя, продемонстрирую несколько сценариев использования ПО:

{гиф сценарий включение лампы}

{гиф сценарий включение всего по появлению устройства в сети}

{гиф сценарий следующий трек}

**Вкратце:**

1. Основная часть ПО - сервер, который крутится на ОС Windows, написан на C#;
2. Клиент на ОС Android, написан на Java;
3. Планируются клиенты на UWP и IOS, web-клиент.

**Некоторые особенности ПО**:

1. Конструктор сценариев. Создание сложных сценариев с циклами и условиями;
2. Работа с устройствами ZWave и Modbus;
3. Вложенность сценариев. Можно один сценарий использовать в другом в качестве процедуры;
4. Запуск удаленного сценария. В сценарии текущего сервера можно запускать сценарий удаленного;
5. Возможность добавлять в систему свою функциональность посредством создания модулей;
6. Запуск сценариев с помощью смартфона.

**Конструктор сценариев**

Программа позволяет создавать как сложный сценарий, так и одиночное действие.

*Одиночное действие* - это сценарий, который содержит только одно действие. В нем нет возможности создавать циклы и условия, а можно лишь “замапить” определенное действие на кнопку в UI клиента.

В качестве примера приведу свой способ их использования: создал категорию “Розетки”, в ней собрал все лампы, бытовой вентилятор, светильники, телевизор и компьютер. Теперь, чтобы включить одно из этих устройств, следует открыть клиент на андроиде, зайти в категорию “Розетки” и выбрать соответствующий пункт меню.

Обычно, одиночные действия использовать неудобно, так как различные устройства удобно запускать при каком-то условии или хотя-бы не по одному. Именно для этого нужен “сложный сценарий”, о чем далее.

*Конструктор сложного сценария* позволяет создавать сценарии с циклами и условиями. Состоит из двух частей: непосредственно сценарий и его завершение.

{скриншот}

*ЕСЛИ и ИНАЧЕ в конструкторе***.** В данных конструкциях можно создавать цепочки условий с логическими операторами НЕ, ИЛИ, И. Также, можно создавать группу условий и использовать операторы непосредственно с группой. Если условие остается пустым, то оно автоматом вычисляется как ЛОЖЬ.

Есть некоторое количество встроенных проверок (такие как проверка на дату, время и т.д.), а так же несколько в качестве подключаемых модулей.

{скриншот}

*Цикл ПОКА в конструкторе.* Цикл состоит из условия и тела. Условие создается аналогично тому, как он создается в конструкции ЕСЛИ (группы условий, операторы НЕ, ИЛИ, И).

{скриншот}

Так же можно настроить следующие параметры сценариев:

-Запуск извне (вкл/выкл), позволяет запершить или разрешить запуск сценария с другой машины или из клиента;

-Автоматический запуск сценария при старте системы;

-Выбор категории сценария, позволяет отображать в UI клиента пункт меню для запуска сценария в определенной категории.

**Работа с устройствами ZWave и Modbus**

На данный момент система поддерживает устройства ZWave (на основе openzwave) и Modbus RTU (на основе nmodbus).

*ZWave* - это беспроводной протокол связи, использующийся в домашней автоматизации. Для реализации протокола используются миниатюрные маломощные радиочастотные модули. Сейчас на рынке ZWave представлен довольно обширный перечень устройств, такие как ZWave лампочки, розетки, релейные модули, переключатели, замки, датчики температуры (влажности, движения, протечки), терморегуляторы, устройства управления кондиционером и т.д. Так же распространены мультисенсоры - устройства, сочетающие несколько датчиков (например датчик освещенности, присутствия, температуры и дверной датчик), что весьма привлекательно с точки зрения финансовых затрат и расположения в помещении.

*Modbus -* это проводной протокол, применяемый для связи между электронными устройствами, которые его поддерживают. В контексте нашей программы может быть полезен для работы с релейными модулями, для чего я его и использую у себя дома. Не буду подробно останавливаться на нем.

**Вложенность сценариев**

Часто бывают такие ситуации, когда одно действие участвует во многих сценариях (например, отключение всех бытовых устройств может происходить по нажатию на пункт меню в программе, по сценарию выхода всех wi-fi устройств из сети, просто по таймеру), и поэтому каждый раз создавать (или править) одинаковый алгоритм в каждом из сценариев не представляется удобным. Для этого существует такое встроенное действие, которое позволяет запускать уже созданный сценарий в текущем. Так же это может быть удобно, когда часто добавляются новые бытовые устройства (или другие частые изменения в алгоритме работы “умного дома”), для изменения работы всех сценариев достаточно будет изменить лишь один сценарий, который используется в них.

{скриншот}

**Запуск удаленных сценариев**

Посредством запуска удаленных сценариев можно запускать сценарий одного сервера в сценарии другого. В качестве примера хочу привести мой вариант использования этого функционала: создал сценарий (на домашнем сервере) под называется “свет+мультимедиа”, который включает свет в помещении, запускает мой десктоп используя WakeOnLan, включает тв, к которому подключен десктоп и ждет пока на нем (десктопе) запустится экземпляр нашей программы, затем запускает на ней сценарии “включить музыку” и “звук на 20 единиц” с помощью запуска удаленных сценариев.

{гиф}

**Написание собственных модулей**

Помимо того, что пользователь может создавать сложные сценарии штатными средствами, программа позволяет “расширять” ее пользовательскими модулями на C#. Все элементы действий и проверок в сценариях (такие как “Проверка по дате”, “Показать сообщение”, “Действие Modbus”, “Действие ZWave” и т.п.) это классы, унаследованные от ICustomAction и ICustomChecker. Следуя определенным правилам, любой пользователь может создать свой модуль, который может быть очень простым (например, озвучивание текущей температуры воздуха за окном), так и сложным, за которым может крыться целый фреймворк. Достаточно наследоваться от интерфейсов ICustomAction или ICustomChecker, расставить несколько атрибутов, скомпилировать DLL и добавить в программу через вкладку “МОДУЛИ” (полная инструкция создания модулей <http://noant.github.io/Pyrite/PyriteMods.htm>).

{скриншот}

**Запуск сценариев с помощью смартфона (пока только OS Android)**

{скриншот экрана смарфтона}

Сценарии отображаются на главном экране и в категориях. Категория является виртуальной “папкой”. При запуске сценария обновляется статус соответствующей кнопки.

В дальнейшем планируется клиент для UWP, IOS, web, а так же модуль для управления устройствами (планы на ближайшее будущее) и создания сценариев с помощью голоса (стратегические планы).

Страничка программы: http://noant.github.io/Pyrite/

Ссылка для скачивания: <http://noant.github.io/Pyrite/downloads.html>

Инструкция для пользования: <http://noant.github.io/Pyrite/description.html>

Инструкция для создания собственных модулей: <http://noant.github.io/Pyrite/PyriteMods.htm>

P.S.: Если нашли ошибку в программе, то пожалуйста, отправляйте ее на [noant1@yandex.ru](mailto:noant1@yandex.ru).