РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ УДАЛЁННОГО УПРАВЛЕНИЯ СИСТЕМАМИ ОСВЕЩЕНИЯ

Никита ЧУКЛЯ

The article covers software aspects of creating a software and hardware system for remote control of lightning systems. The main thesis is implementing extensible data access and remote-control protocol which is used by both control and managed sides. This extensible protocol is based on all present capabilities of manageable devices pool, but takes into account possible limitations of each device.

В настоящее время системы автоматизации становятся всё более популярными, что обусловлено снижением их стоимости, расширением функционала и повышением информационной грамотности населения. Современные системы автоматизации основываются на концепции «интернета вещей», значительно расширяющей возможности анализа, сбора и распределения информации, на основе которой и принимаются дальнейшие решения [1]. Одной из областей применения систем автоматизации является область удалённого контроля систем освещения.

Современные комплексы управления системами освещения могут включать в себя большое количество компонентов и реализовывать в себе поведение множества отдельных устройств, таких как умные выключатели, умные диммеры, датчики движения, присутствия, открытия дверей и окон, датчики света, таймеры, интеграцию с внешними системами (охранной, пожарной, предупреждения о бедствиях) [2]. Все компоненты таких комплексов тесно связаны между собой логически и зависимы друг от друга. В зависимости от реализации самой системы, может быть добавлена также частичная или полная автоматизация сценариев управления освещением с удалённым контролем лишь логических параметров, используемых в таких сценариях [3].

Исходя из того факта, что возможности системы могут включать в себя большое количество функций и управляемых параметров, а также того, что каждая отдельно взятая система не включает

в себя все из этих возможностей, возникает необходимость в создании протокола сообщения между управляющими и управляемыми устройствами, который учитывал бы данные особенности и позволял получать от управляемого устройства список его функций и то, как можно ими воспользоваться.

```
{
  "type": "POWER_CONTROL",
  "input": [
    {
      "type": "POWER_OFF",
      "description": "Turn power off",
      "endpoint": "/power",
      "value": 0
    },
    {
      "type": "POWER_ON",
      "description": "Turn power on",
      "endpoint": "/power",
      "value": 1
    }
  ],
  "output": [
    {
      "type": "POWER_STATUS",
      "endpoint": "/status",
      "field": "powerStatus"
    }
  ]
}
```

Рис. 1. Описание способности устройства управлять питанием в формате JSON

Предложенный ниже подход является абстрактной концепцией для создания протокола подобного типа. Основополагающей его частью является возможность получить список возможностей устройства при отправке запроса по заранее установленному пути, который может иметь вид *адрес_устройства/**capabilities*** и затем использовать полученную информацию для работы с этими возможностями. В ответе, содержащем список возможностей, который в случае предлагаемой реализации представляет собой массив JSON, каждая из возможностей устройства описывает саму себя и позволяет запросившему список управляющему устройству определить, по каким путям обращаться для получения информации либо запуска функционала управляемого устройства.

Примером может послужить управляемое устройство, одной из возможностей которого является управление подачей питания. Для устройств, поддерживающих такой функционал, описание способности представлено на Рисунке 1.

Каждая из возможностей обладает собственным типом, в примере выше её тип – «POWER_CONTROL». Массив input состоит из объектов, каждый из которых содержит информацию об атомарной операции описываемой возможности. В данном случае можно осуществить операции двух типов – «POWER_OFF» и «POWER_ON», то есть выключение и включение питания нагрузки. Полученная из такого JSON-объекта информация также содержит в себе путь, по которому нужно совершить запрос, значение параметра, с которым нужно совершить запрос, а также описание действия.

Массив output содержит в себе объекты, описывающие способы получения связанной с возможностью данного устройства информации. В данном случае полезно знать, в каком состоянии находится питание нагрузки в данный момент, прежде чем его изменять. Объекты этого массива содержат тип и описание, а также путь, обратившись по которому можно получить релевантную информацию, и поле из возвращаемого JSON-объекта, значение которого соответствует этой информации.

Такой подход позволяет реализовать динамически адаптируемый интерфейс со стороны управляющей части, показывая пользователю только те элементы управления, которые доступны для выбранного устройства. Пример такого интерфейса приведен ниже (Рис. 2).

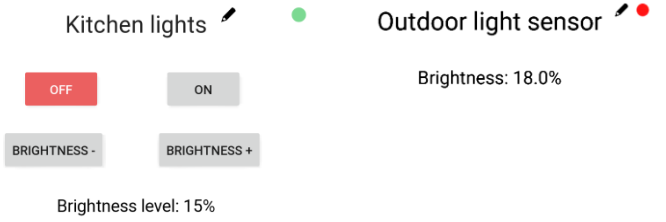


Рис. 2. Реализация динамически адаптируемого интерфейса устройства управления

Благодаря реализации расширяемого протокола сообщения и получения данных о возможностях устройства со всей информацией для их использования, пользователь может не только получать релевантную информацию о текущем состоянии этого устройства, но и управлять им с помощью удобного интерфейса, не содержащего элементов управления, которые не соответствовали бы возможностям отдельно выделенного устройства.

Литература:

1. CHAUHURI, A. *Internet of Things, for Things, and by Things*. USA, Florida: CRC Press, 2018, pp. 78-79.
2. ROWLAND, C., GOODMAN, E., CHARLIER, M., LIGHT, A., LUI, A. *Designing Connected Products: UX for the Consumer Internet of Things*. USA, California, O'Reilly Media, 2015, p. 29.
3. DILOUIE, C. *Lighting Controls Handbook*. USA, Georgia, Fairmont Press, 2008, pp. 14-17.

Рекомендовано
Денис НИКА, др. хаб.