

UDC 621.391.8

Sidorkovich D.R., Fedorov A.S. Node-RED software development for access and control system RFID-based

Разработка программного обеспечения в среде Node-RED для системы контроля и управления доступом на базе технологии радиочастотной идентификации

Sidorkovich Daria Romanovna,

Graduate Student of Department of Radio Communication and Broadcasting, SPbSUT

Fedorov Andrey Sergeevich,

Engineer of Research and Education Center «Wireless Infotelecommunication Networks», SPbSUT

Scientific adviser: **Vorobiev O.V.**, Cand. Of Eng. Sc., professor, head of Department of Radio Communication and Broadcasting, SPbSUT

Сидоркович Дарья Романовна,

Магистрант кафедры радиосвязи и вещания, СПбГУТ им. проф. М.А. Бонч-Бруевича
Федоров Андрей Сергеевич,

Инженер научно-образовательного центра «Беспроводные инфотелекоммуникационные сети», СПбГУТ им. проф. М.А. Бонч-Бруевича

Научный руководитель

Воробьев О.В., к.т.н., профессор, заведующей кафедрой радиосвязи и вещания, СПбГУТ им. проф. М.А. Бонч-Бруевича

***Abstract.** This article focuses on the development of access and control system software in the Node-RED cloud graphics programming environment. The system is based on RFID-equipment manufactured by Advantech. A link to the results of the previously developed prototype is presented, as well as a description of the changes made to its work, fragments of the source text of the program and examples of work.*

***Keywords:** access and control system, RFID, Node-RED, Advantech.*

***Аннотация.** Данная статья посвящена разработке программного обеспечения системы контроля и управления доступом в облачной графической среде программирования Node-RED. Система основана на оборудовании, работающем на основе технологии радиочастотной идентификации и произведенном компанией Advantech. Представлена ссылка на результаты ранее разработанного прототипа, а также описание внесенных в его работу изменений, фрагменты исходного текста программы и примеры работы.*

***Ключевые слова:** СКУД, радиочастотная идентификация, Node-RED, Advantech.*

DOI 10.54092/25421085_2021_11_99

Рецензент: Сагитов Рамиль Фаргатович, кандидат технических наук, доцент, заместитель директора по научной работе в ООО «Научно-исследовательский и проектный институт экологических проблем», г. Оренбург

Данная статья продолжает работу, описанную в [1] и посвящена повышению эффективности работы программной части разработанной СКУД, путем ее оптимизации и добавления новых функций.

Аппаратная часть системы реализована на основе базовой станции технологии RFID модели WISE-2834 компании Advantech. Это универсальное устройство, имеющее возможность быть интегрированным в локальную сеть, развернутую на объекте, посредством подключения по Ethernet-интерфейсу, а также имеющее возможность подключения внешних устройств и датчиков к встроенным цифровым входам и выходам. Внешний вид устройства и основные характеристики представлены на рисунках 1 и 2 соответственно [2].



Рисунок 1. WISE-2834

| Параметр | Значение |
|---|---|
| Поддерживаемые частотные диапазоны, МГц | US: 902.75–927.25 EU: 865.7–867.5 TW: 922.25–927.75 |
| Мощность передачи, дБм | 10–31.5 |
| Чувствительность, дБм | –82 |
| Чипсет | ARM Cortex-A8 300 MHz |
| Оперативная память, Мб | 512 DDR3L |
| Объем памяти, Мб | 512 |
| Интерфейсы | Status, Serial (Tx, Rx), Wi-Fi communication, RFID Channel on/off, Wi-Fi Signal Strength, 1 x Micro SD card, 1 x USB2.0 High Speed, 1 x 10/100 Based-T RJ-45, 1 x RS-485, 4 Digital Inputs, 4 Digital Outputs |
| Напряжение электропитания, В | 10–30 DC |
| Габариты, мм | 190×120×30.2 |
| Рабочая температура, °С | –25 ~ +50 |
| Рабочая влажность, % | 20 ~ 95 |

Рисунок 2. Основные характеристики

Программная часть выполнена в облачной графической среде программирования Node-RED [3] и, по сравнению с предыдущей итерацией, несколько усложнена. Node-Red является оптимальным решением, так как СКУД разворачивается на небольшой территории, а персонал, которому нужно предоставить доступ, немногочислен. Весь инструментал, предоставленный программой, позволит объединить все элементы СКУД в единую систему, также создать необходимую базу данных, на основании которой будет предоставляться доступ сотрудникам в выбранные помещения.

Разработанная программа подчиняется логике, описанной блок-схемой, представленной на рисунке 3.

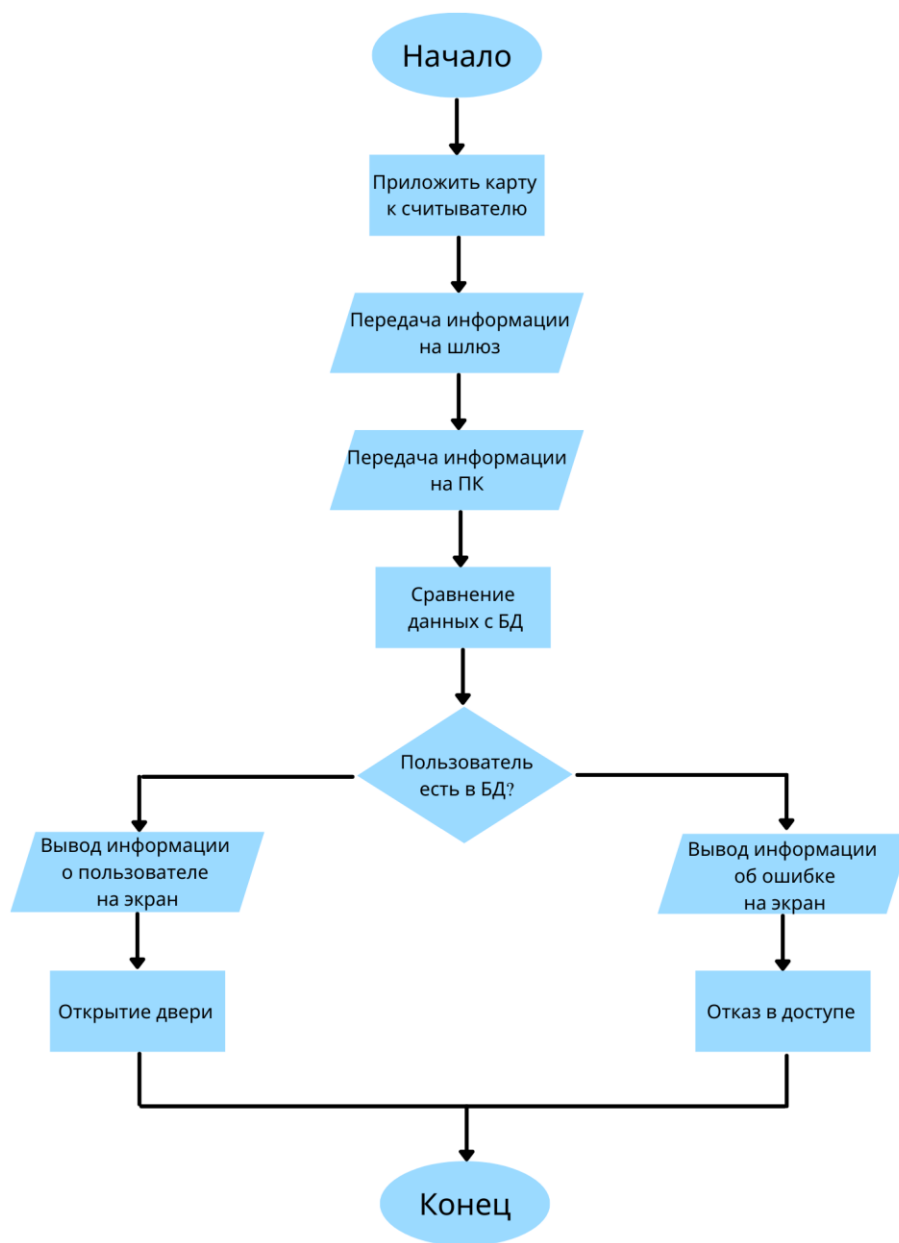


Рисунок 3. Блок-схема программы

Программа, разработанная в среде Node-RED, в графическом виде представлена на рисунке 4.

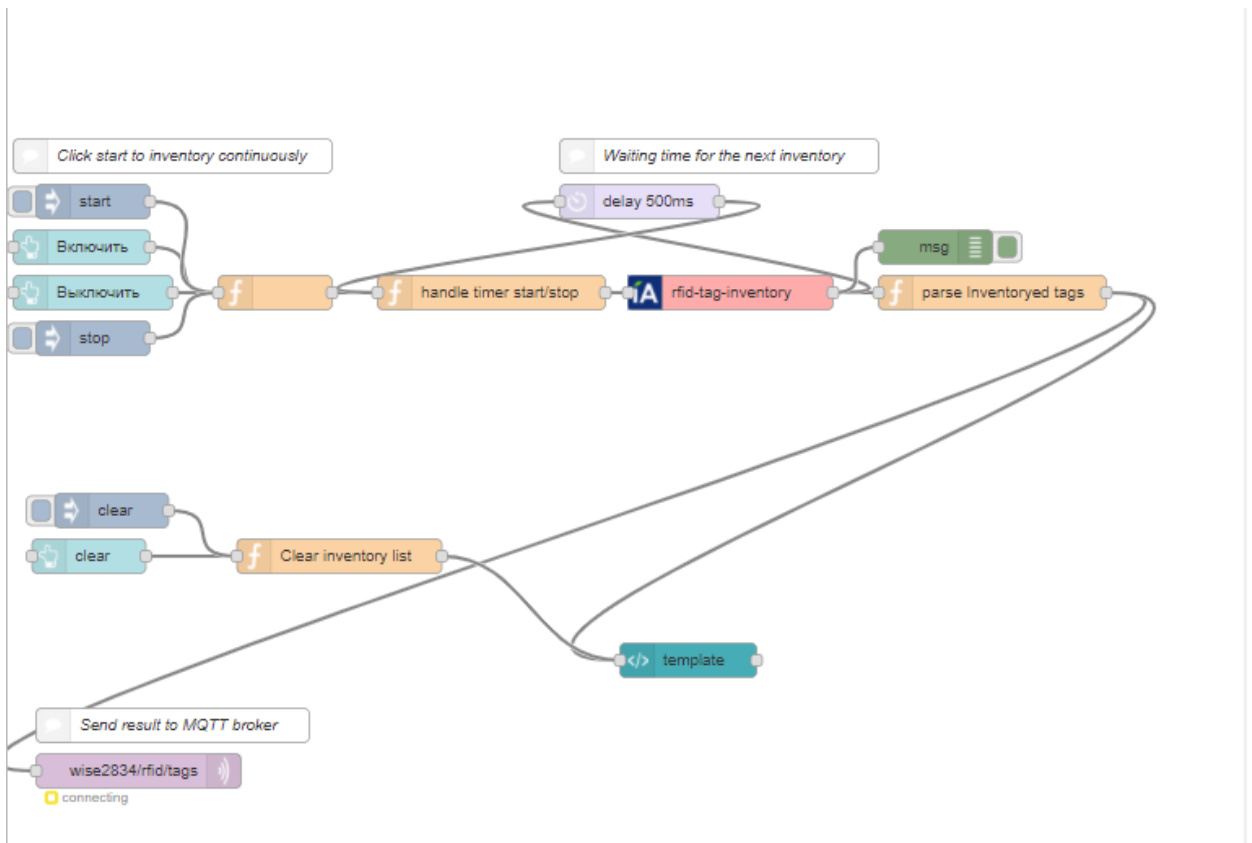


Рисунок 4. Программа в среде Node-RED

Каждый отдельный блок имеет свое наполнение. Для ввода сообщения в поток данных используются ноды «start» и «stop». Они активируются после нажатия кнопок включения и выключения. После получения сигнала о начале работы, сообщение передается в функции, которые обрабатывают информацию. Данная функция описывает процесс начала и остановки обработки сообщения. Ниже представлено внутренне наполнение первой используемой функции, которая привязана к узлам «start» и «stop»:

```
var inv_state == global.get ('inv_state') || 'stop';
if (inv_state == 'start')
    {
    Var New={};
    New.payload='go';
    return New;
    }
else
{
}
```

Далее используется узел, непосредственно относящийся к устройству – «rfid-tag-inventory». Он позволяет хосту собирать Electronic Product Code (EPC) для

всех интересующих тегов, которые будут входить в контакт с исследуемой системой.

Для того, чтобы избежать наложения информации от разных носителей, используется функция задержки «delay», в данном случае она равна 500 мс. Это необходимо для предотвращения потери приходящих сообщений, которые могут быть повреждены избытком информации.

После получения сообщения, функция «parse inventory tag» разбирает его на нужную нам информацию. Ниже представлен внутренний код данной ноды, который осуществляет необходимые нам действия:

```
var ary = [];  
var hit = false;  
for (var i=0;i<msg.Inv.length;i++)  
{  
  hit = false;  
  var tempEPC = (msg.Inv[i].acc_data).toString('hex');  
  for (var j=0;j<ary.length;j++)  
  {  
    if (ary[j].epc == tempEPC)  
    {  
      hit = true;  
      ary[j].cnt++;  
      //node.error("j="+j+" cnt="+ary[j].cnt);  
      if (msg.Inv[i].logic_ant == 0)  
        ary[j].ant0++;  
      if (msg.Inv[i].logic_ant == 1)  
        ary[j].ant1++;  
      if (msg.Inv[i].logic_ant == 2)  
        ary[j].ant2++;  
      if (msg.Inv[i].logic_ant == 3)  
        ary[j].ant3++;  
    }  
  }  
}  
if (hit === false)  
{  
  var obj = new Object();  
  obj.epc= tempEPC;  
  obj.cnt = 1;  
  obj.ant0 = 0;  
  obj.ant1 = 0;  
  obj.ant2 = 0;  
  obj.ant3 = 0;
```

```

    if (msg.Inv[i].logic_ant == 0)
        obj.ant0++;
    if (msg.Inv[i].logic_ant == 1)
        obj.ant1++;
    if (msg.Inv[i].logic_ant == 2)
        obj.ant2++;
    if (msg.Inv[i].logic_ant == 3)
        obj.ant3++;

    obj.time = new Date().toISOString();
    ary.push(obj);

}
var InventoryList = global.get('InventoryList')||[];
InventoryList = ary;
global.set("InventoryList",InventoryList);
msg.payload = InventoryList;

return msg;

```

После разбора информации на отдельные пункты, поток данных передается в созданную таблицу, она может быть представлена в графическом виде. Для этого действия используется узел «template», внутренний код которой продемонстрирован ниже:

```

<table style="width:100%">
  <tr>
    <td>Номер</td>
    <td>ФИО</td>
    <td>Должность</td>

    <td>Аудитория 1</td>
    <td>Аудитория 2</td>
    <td>Аудитория 3</td>
    <td>Аудитория 4</td>
    <td>Время</td>

  </tr>
  <tr ng-repeat="x in msg.payload ">
    <td>{{$index}}</td>
    <td>{{msg.payload[$index].epc}}</td>
    <td>{{msg.payload[$index].cnt}}</td>

```

```

<td>{{msg.payload[$index].ant0}}</td>
<td>{{msg.payload[$index].ant1}}</td>
<td>{{msg.payload[$index].ant2}}</td>
<td>{{msg.payload[$index].ant3}}</td>
<td>{{msg.payload[$index].time}}</td>
</tr>
</table>

```

Информация, полученная благодаря узлу «parse inventory tag», представляется в удобном для пользователя виде. Также для сброса уже полученной информации, оформленной в таблицу, используется функция «Clear».

Полученные результаты можно отправить в MQTT-брокер. Он используется для общения между умными устройствами, путем пересылки сообщений о своем состоянии. Для этого используется Wi-Fi сеть и некое центральное устройство под названием брокер. В его задачу входит слушать сеть, получать и передавать полученные сообщения и хранить их, если это запрошено.

Самым важным нововведением в программе стала возможность вывода на экран (например, это может быть компьютер дежурного сотрудника) описанной выше таблицы с указанием аудитории, куда была предпринята попытка доступа, ФИО посетителя, степень его допуска (в зависимости от должности), а также время и дата попытки доступа. Пример этого экрана приведен на рисунке 5.

| ВКЛЮЧИТЬ | | ВЫКЛЮЧИТЬ | | CLEAR | | | |
|----------|---------------------------------|---------------|-------------|-------------|-------------|-------------|--------------------------|
| Номер | ФИО | Должность | Аудитория 1 | Аудитория 2 | Аудитория 3 | Аудитория 4 | Время |
| 0 | Сидоркович Дарья Романовна | Студент | 1 | 0 | 0 | 0 | 2021-06-04T14:09:23.910Z |
| 1 | ERROR! Должность не установлена | | 1 | 0 | 0 | 0 | 2021-06-04T14:07:15.673Z |
| 2 | Иванов Иван Иванович | Преподаватель | 1 | 0 | 0 | 0 | 2021-06-04T14:11:59.232Z |

Рисунок 5. Пример работы программы

Таким образом, процесс работы системы можно описать следующими шагами:

Процесс доступа является многоступенчатым и завязан непосредственно на автономной работе шлюза и компьютера. Его можно описать следующим образом:

1. Пользователь прикладывает тег с уникальным номером к считывателю, который находится непосредственно около входа в аудиторию. В зависимости от дальности считывания антенны, существует возможность настройки системы, при которой нет необходимости прикладывать метку, она может считываться при

- проходе через дверной проем. Этот способ удобен для тех помещений, где нужно фиксировать время входа и выхода персонала.
2. Считыватель передает информация на RFID - шлюз, в нашем случае Wise - 2834. Сам шлюз крепится в непосредственной близости от антенны на любой удобной поверхности.
 3. Полученная информация отправляется на компьютер.
 4. Система сравнивает номер метки с данными, находящимися в базе. В таблице 1 приведены данные, используемые при пуско-наладочных работах системы.

Таблица 1

Исходные данные для рассматриваемого примера

| Тег | ФИО | Должность |
|----------------------------------|-----------------------------|---------------|
| 3000e200001d52100102108045e7325c | Сидоркович Дарья Романовна | Студент |
| 3000e2006000621201360600d7200717 | Иванов Иван Иванович | Преподаватель |
| 3000e2006000621201900830c2270b68 | Петрова Ирина Александровна | Студент |

5. При совпадении разрешается вход в аудиторию и на экране появляются все необходимые сведения о пользователе.
6. В случае, если система не обнаруживает совпадений, на экране появляется ошибка и дверь в аудиторию блокируется.

Система начинает работу при нажатии кнопки «ВКЛЮЧИТЬ». После этого запускается процесс, описанный выше. Чтобы остановить считывание необходимо нажать «ВЫКЛЮЧИТЬ». После прекращения работы все данные остаются. Для очистки поля нажимается «CLEAR», и оно приходит в исходный вид. Срабатывание системы обозначается двумя уровнями: «1» и «0», где 1 – антенна была активирована и был запрошен доступ в определенную аудиторию, а 0 – система находится в состоянии покоя.

Исследование выполнено в рамках выполнения ПНИ по государственному заданию СПбГУТ на 2021 год.

References

1. А. А. Прасолов, Д. Р. Сидоркович, А. С. Федоров. Разработка СКУД на базе технологии RFID // АПИНО 2021. Сборник научных статей. ТОМ 3. Радиотехнологии связи. Цифровая экономика и управление в связи. Гуманитарные проблемы информационного пространства.
2. WISE-2834. Intelligent RFID Gateway. Datasheet. URL: [https://advdownload.advantech.com/productfile/PIS/WISE-2834/file/WISE-2834_DS\(012721\)20210127141714.pdf](https://advdownload.advantech.com/productfile/PIS/WISE-2834/file/WISE-2834_DS(012721)20210127141714.pdf).
3. Node-RED – Low-code programming for event-driven application. URL: <https://nodered.org/>