

Автоматизация предприятий общественного питания с использованием современных цифровых технологий: программ Arduino IDE, OPC Modbus и Master Scada

Владимир В. Русанов ¹	kafedrates@mail.ru
Виталий И. Перов ¹	perov-vitaliy@bk.ru
Михаил А. Самойлов ¹	Mi1995@yandex.ru

¹ Российский Экономический Университет им. Г.В. Плеханова, Стремянный пер., 36, г. Москва, 117997, Россия

Реферат. В статье рассмотрено пошаговое действие по использованию трех компьютерных программ: «Arduino IDE», «Modbus OPC Universal Modbus Server» и «Master Scada» для создания проектов автоматизации предприятий общественного питания. Фактическое повторение приведенного в работе примера автоматизации горячего цеха позволяет освоить методы проектирования на базе современных цифровых технологий. В представленных методах автоматизации для измерения параметров технологических процессов производства блюд на предприятиях общественного питания мы рекомендуем использовать интегральные датчики. Существующие датчики температуры, давления, перемещения, влажности хорошо адаптируются с современными микроконтроллерами, что позволяет создавать компактные недорогие устройства первичного сбора и обработки информации. На примере созданного в работе OPC-сервера показан метод передачи собранной информации в компьютер диспетчера, следящего за процессами варки продуктов в горячем цеху предприятий общественного питания. Диспетчерский пульт управления процессом разработан на базе программы «Master Scada». Это одна из наиболее широко используемых для автоматизации программ, где разработчики указывают на ее применение в таких компаниях как «Газпром» и «Росатом». Широкие возможности программы позволяют в трехмерной графике представлять объекты автоматизации, динамизировать исполнительные и регулирующие механизмы, находящиеся на экране перед диспетчером. Данный в работе практический пример ограничен тремя мнемосхемами, первая из которых является заставкой горячего цеха с меню перехода на мнемосхему управления пищеварочным котлом и электрическим жарочным шкафом. При переходе на мнемосхему пищеварочного котла оператор (диспетчер) видит степень заполнения котла варочной жидкостью, температуру этой жидкости, время варки. С помощью кнопок управления оператор способен внести изменения в программу приготовления своего блюда. Также программа следит за аварийными режимами, в частности за недопущением сухого хода.

Ключевые слова: микроконтроллер, OPC сервер, мнемосхема, пищеварочный котел.

Automation of public catering enterprises using modern digital technologies: Arduino IDE, OPC Modbus and Master Scada programs

Vladimir V. Rusanov ¹	kafedrates@mail.ru
Vitalii I. Perov ¹	perov-vitaliy@bk.ru
Mikhail A. Samoilov ¹	Mi1995@yandex.ru

¹ Plekhanov Russian University of Economics, Stremyanny per., 36, Moscow, 11997, Russia

Summary. In the article, the using of three computer programs was considered step-by-step. These programs are "Arduino IDE", "Modbus OPC Universal Modbus Server" and "Master Scada". They are using for creating projects of automation for public catering enterprises. There is an example of automation of the hot shop in this article, which demonstrates us the methods of engineering based on the modern digital technologies. If you want to learn how to create the projects using these programs, for the beginning, we recommend you to repeat our demonstration project. In the represented methods, we recommend using the integrated sensors for measuring options of technological processes of production dishes. Modern sensors of temperature, pressure, moving and humidity are adapting with microcontrollers very good. This fact allows to create compact, inexpensive devices for primary collection and processing of information. There has been showed the method of transferring of collected information to the computer of the supervisor on the example of created OPC-server. The supervisor can look for the processes, which are happening in the hot shop of the public catering establishment. The supervisor's remote control is developed on the base of the «Master Scada» program. This is one of the most popular automation program. Creators of the «Master Scada» say that this program is widely used in such companies as «Gazprom» and «Rosatom». The wide possibilities of this program allow us in the three-dimensional graph to represent the objects of automation, to accelerate the executive and regulating mechanisms that are on the screen in front of the dispatcher. The practical example, which was demonstrated in this article, is limited by three mnemonic diagrams. First of them is the splash screen of the hot shop with the menu of the transitioning to the mnemonic diagram of the controlling of the food-brewing boiler and the electrical frying cupboard. If operator switches over to mnemonic diagram of the food-brewing boiler, he sees the degree of filling boiler by the cooking liquor, the temperature of this liquor and time of the cooking. Using the control buttons operator can make changes to the program of cooking his dish. The program also looks for emergency modes, in particular, to prevent dry running.

Keywords: microcontroller, OPC server, mnemonic diagram, digestion boiler.

Для цитирования

Русанов В.В., Перов В.И., Самойлов М.А. Автоматизация предприятий общественного питания с использованием современных цифровых технологий: программ Arduino IDE, OPC Modbus и Master Scada // Вестник ВГУИТ. 2018. Т. 80. № 2. С. 38–44. doi:10.20914/2310-1202-2018-2-38-44

For citation

Rusanov V.V., Perov V.I., Samoylov M.A. Automation of public catering enterprises using modern digital technologies: Arduino IDE, OPC Modbus and Master Scada programs. *Vestnik VGUET* [Proceedings of VSUET]. 2018. vol. 80. no. 2. pp. 38–44. (in Russian). doi:10.20914/2310-1202-2018-2-38-44

Введение

Три компьютерные программы, перечисленные в заголовке статьи, позволяют разрабатывать проекты автоматизации технологических процессов различной сложности.

Arduino IDE – предназначена для создания управляющих программ и прошивки микроконтроллеров. Различная производительность выпускаемых микроконтроллерных плат, от arduinonano до arduinomega, позволяет выбрать оптимальный вариант экономической целесообразности при планировании создаваемого проекта [1].

OPCModbus – позволяет создать сервер передачи информационных сигналов от микроконтроллерной платы на диспетчерский компьютер.

MasterScada – как любая другая skada, предназначена для приема сигналов от контроллеров и создания мнемосхем на пунктах диспетчеризации.

Из рекламных роликов разработчика, MasterScada использовалась в проектах Газпрома, при автоматизации атомных электростанций, нефтяных платформ и т. п.

Результаты и обсуждение

Как создается проект, рассмотрим на конкретном примере автоматизации горячего цеха предприятия общественного питания.

Первый шаг, это разработка плана проекта. Здесь отмечаются объекты автоматизации, подбираются датчики определения параметров на всех этапах технологического процесса, рассматриваются исполнительные механизмы и устройства, которыми осуществляется процесс. В нашем проекте технолог (шеф повар), следит за приготовлением пищи в варочном котле, сидя перед диспетчерским пультом управления.

Он должен видеть:

А) Общий вид горячего цеха;

Б) Выбор (меню) любого оборудования для определения его состояния;

Г) При открытии мнемосхемы конкретного устройства (котла или плиты) иметь представление о состоянии объекта и параметрах идущего там процесса.

В выбранном нами варочном котле готовятся бульоны, первые блюда, напитки, проводится кипячение молока.

Как правило, вначале, жидкость, помещенная в котел, доводится до кипения, а затем определенное время идет варка при пониженной мощности.

Следовательно, необходимы датчики для определения уровня жидкости и температуры

продукта в варочном сосуде, часы для определения времени, устройство отключения котла в аварийных ситуациях (сухой ход, отсутствие продукта в варочном пространстве).

Второй шаг – создание и проверка работы программы управления на микроконтроллерной плате (Используем программу ArduinoIDE).

Перед началом работы необходимо скопировать папки DallasTemperature и SimpleModbusSlaveV9 в папку arduino/libraries. Эти две папки являются библиотеками для работы arduino с датчиком температуры DS18B20 [2] и программой Modbus Universal MasterOPC Server. Далее выполняем по пунктам:

А) Выбираем конкретную микроконтроллерную плату, например – «arduinouno».

Б) Подключаем ее к компьютеру и запускаем программу «Arduino IDE»

В) Создаем управляющую программу и прожигаем ПЗУ микроконтроллера.

Г) Подключаем к микроконтроллеру датчики измерения параметров и приборы, моделирующие работу исполнительных устройств котла.

Д) Проверяем работоспособность, устраняем недостатки.

На этом шаге желательно вначале разработать алгоритм управляющей программы. Для нашего случая управления варочным котлом порядок работы алгоритма следующий:

1. После включения микроконтроллера проходит загрузка начальных данных (задаются константы и имена переменных, подключаются необходимые библиотеки для обслуживания датчиков и индикаторных устройств пульта управления).

2. Измеряются параметры (отсутствие сухого хода, наличие жидкости в варочном сосуде).

3. В результате измерений проводится ветвление (либо выдача аварийного сигнала, либо переход на нагрев жидкости до кипения).

4. Измеряется температура и объем жидкости, в результате измерений диспетчером выдаются команды на добавление жидкости или перехода на следующие этапы варки.

5. При достижении температуры кипения происходит переключение мощности и, диспетчер по часам программы выбирает время варки.

Если создаем программу управления работой всего оборудования горячего цеха, в нее добавляется код каждого агрегата и алгоритм его управления, подобный рассматриваемому варианту варочного котла.

Скетч для управления котлом составленный нами по данному алгоритму представлен ниже.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
#define TEMPERATURE_PRECISION 9
int senPin = A0; // устанавливаем вход-
ную ногу для АЦП
int senValue = 0; // начальное цифро-
вое значение
int RELE = 13; // Выбираем выход ава-
рийного сигнала
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sen-
sors(&oneWire); //Первый sensor
подключен к библиотеке
float temp1 = 0; // переменная для
температуры первого датчика
DeviceAddress Thermometer1 = {
0 x28, 0xFF, 0 x09, 0 x16, 0 x63,
0 x04, 0 x00, 0xF0}; // адрес датчика
температуры
void setup()
{
pinMode(RELE, OUTPUT); // настраиваем
выход аварийного отключения
pinMode(5, INPUT); // настраиваем 5
вход сухого хода
sen-
sors.begin(9600); //настройка монитора
sensors.setResolution(9);
// для 3 датчиков
}
void loop()
{
boolean сухой = digitalRead(5); // за-
носим значение сухого хода
if(сухой = true){
tone (3, 100, 5000); // если авария
лог1 включаем сирену
}
Serial.println (сухой);
senValue = analogRead(senPin); // счи-
тываем значение с объема жидко-
сти delay(10);
if(senValue ≤ 15){
digitalWrite(RELE,HIGH); // включаем
Реле на 13 ноге
}
else{
digitalWrite(RELE,LOW); // выключаем
}
Serial.println (senValue); // выводим
результат измерения объема для про-
верки
sen-
sors.waitForConversion(false); //
Начинаем измерять температуру
sensors.requestTemperatures(); // За-
пуск первого sensor
```

```
sensors.waitForConversion(true);
delay (750);
sensors.getAddress(Thermometer1,
0); // Измерение первым термометром
temp1 = sen-
sors.getTempC(Thermometer1); // Тем-
пература первого датчика
sensors.waitForConversion(false);
Serial.println (temp1);
delay (100);
// Полученные значения передаем в
OPC сервер.
```

После проверки работоспособности скетча и прожига микроконтроллера, подклю-чаем все приборы, имитирующие работу устройства и проверяем действие программы. В данном случае сигнал с датчика температуры мы подаем на второй цифровой вход платы arduino, аналоговый сигнал с датчика уровня жидкости в варочном сосуде на вход A0, цифро-вой логический сигнал защиты от сухого хода на цифровой вход 5. Если в варочном сосуде мало жидкости (высота менее 15 см.), на выходе 13 аварийный сигнал.

На рисунке 1 представлена фотография платы arduino с подключенными датчиками и светодиодными индикаторами визуализации выходных сигналов.

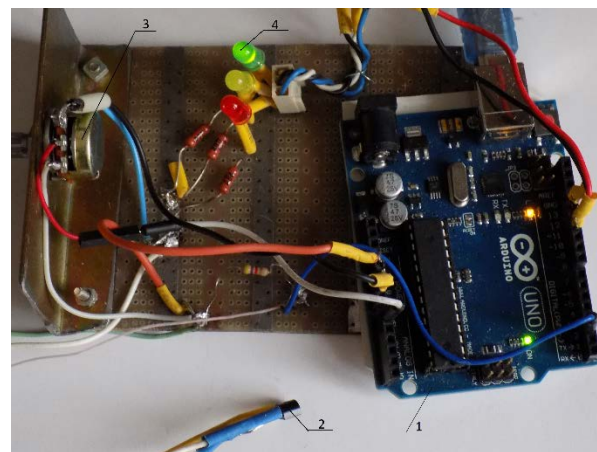


Рисунок 1. Испытание запрограммированной платы: 1 – плата ArduinoUno; 2 – Интегральный датчик температуры DS18B20; 3 – Регулятор аналогового сигнала объема жидкости; 4 – Светодиодные индикаторы состояния исполнительных устройств

Figure 1. The test Board is programmed: 1 – Arduino Uno; 2 – the Integrated temperature sensor DS18B20; 3 – a Regulator of the analog signal of volume of the liquid; 4 – Led status indicators actuators

1. Третий шаг – Передача значений из Arduino в OPC-Server [3].

На компьютер, устанавливаем программу Modbus Universal Master OPC Server.

Если не сделали это раньше, то сейчас библиотеку SimpleModbusSlaveV9 занесите в папку `arduino/libraries`.

Из скетча, созданного на втором шаге, уберите ненужные команды и вместо них введите команды передачи данных в OPC сервер. Пример ниже.

```
#include <SimpleModbusSlave.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
#define TEMPERATURE_PRECISION 9
int senPin = A0; // устанавливаем входную ногу для АЦП
int senValue = 0; // цифровое значение
int RELE = 13;
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire); // Первый sensor
// подключен к библиотеке
float temp1 = 0; // переменная для температуры первого датчика
DeviceAddress Thermometer1 = {
  0 x 28, 0xFF, 0 x 09, 0 x 16, 0 x 63,
  0 x 04, 0 x 00, 0 x F0}; // адрес датчика
1
```

```
const int HOLDING_REGS_SIZE = 3; //
Важно - задаем количество точек передачи в OPC
unsigned int
holdingRegs[HOLDING_REGS_SIZE];
void setup()
{
  pinMode(RELE, OUTPUT);
  sensors.begin();
  sensors.setResolution(9);
  modbus_configure(&Serial, 115200,
    SERIAL_8N2, 1, 2, HOLDING_REGS_SIZE,
    holdingRegs); // вводим скорость опроса
}
```

```
modbus_update_comms(115200,
  SERIAL_8N2, 1);
```

```
// для УЗ трех точек - передаем данные температуры и объема воды
}
void loop()
{
  senValue = analogRead(senPin); // считываем значение объема жидкости
  delay(10);
```

```
if(senValue < 150){
  digitalWrite(RELE, HIGH); // включаем Реле
}
else{
  digitalWrite(RELE, LOW); // выключаем
}
sensors.waitForConversion(false);
```

```
sensors.requestTemperatures(); // Запуск перво-
rosensorsensors.setWaitForConversion(true);
delay(750);
sensors.getAddress(Thermometer1, 0); // Измерение первым термометром
temp1 = sensors.getTempC(Thermometer1); // Температура первого датчика
sensors.waitForConversion(false);
delay(100);
// Полученные значения передаем в OPC
modbus_update();
holdingRegs[1] = senValue; // адрес 1 фоторезистора
holdingRegs[2] = temp1; // адрес 2 DS18D20
}
```

Прошитый вторым скетчем `arduino` подключаем к компьютеру, запускаем программу Modbus Universal Master OPC Server и создаем новую конфигурацию (рисунок 2).

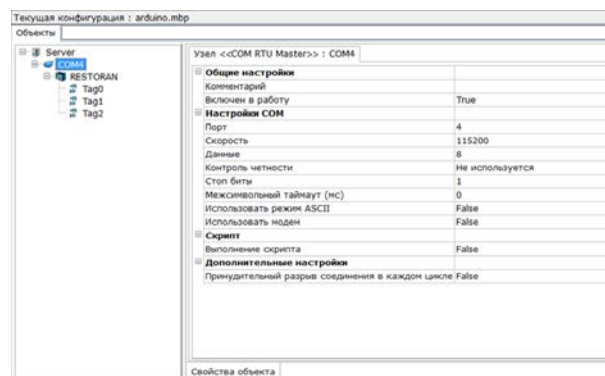


Рисунок 2. Вид программы при создании конфигурации управления через сервер

Figure 2. Type of program when creating a control configuration via the server

Порядок работы при создании конфигурации следующий:

1. После запуска программы выбираем пункт «создать» (из меню).

2. Под окном «Объекты» активизируем имя «Server» и щелкаем на заголовке – **Добавить узел**.

В появившемся окне параметров изменим название узла на COM4. (Порт `com4` использовался при программировании нашего ардуино).

3. Выбираем пункт **Добавить устройство**. Имя устройства «RESTORAN».

В окне параметры вводим нужные – тип узла – COM; порт – 4 (рисунок 3).

4. Выбираем пункт **Добавить tag**. Первый tag – нулевой, следующие tag1 и tag2 – связаны с ячейками (именами) программы ардуино, где хранятся данные измерений температуры и объема жидкости. Сохраняем на диске

созданный файл под именем arduino.mbr. Если Вы сохраняете файл по умолчанию в каталоге «SERVERCFG» можно не концентрировать внимание при дальнейшем его использовании. Если сами назначаете каталог важно запомнить путь к этому файлу, т. к. при создании scada, потребуется связать этот файл и файл, создаваемый в программе MasterScada.

Проверяем проделанную работу. Для этого подключаем к компьютеру плату arduinouno с датчиками измерения температуры и объема воды. Запускаем OPC сервер и открываем файл arduino.mbr. В верхнем левом углу программы щелкаем значок пиктограммы и выбираем «Старт» на экране мы должны получить картину, как на рисунке 3.

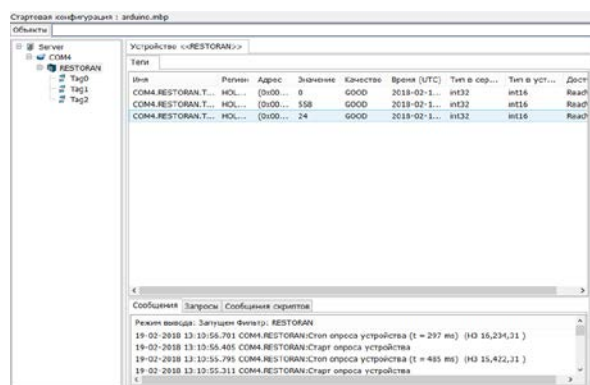


Рисунок 3. Данные из ардуино поступают в OPC сервер

Figure 3. The data from the Arduino go to dis server

В окне результатов «значение» число 558 – это объем воды в варочном сосуде. Максимально возможный объем 1024, ниже температура 24 °C.

Щелкаем кнопку «Стоп» и закрываем программу.

Четвертый шаг – создаем проект в MasterSkada и связь между программами.

Для этого, вначале запускаем созданный нами OPC сервер arduino.mbr. Не начиная измерений, сворачиваем окно этой программы и **запускаем MasterScada**. Выбираем опцию создание нового проекта, даем ему имя и код доступа [4]. Для создания связей между проектом и OPC сервером выполняем следующие действия:

А) В окне дерева системы правой кнопкой мышки щелкаем на слове «Система» рисунок 4.

В выпадающем окне выбираем опцию – «вставить», затем «компьютер».

На панели проектов появляется имя «Компьютер 1».

Б) Правой кнопкой мышки щелкаем на появившемся слове, выбираем вставить «OPC сервер». И в новом окне выбираем «Insat Modbus OPC Server DA». Появятся данные, представленные на рисунке 4.

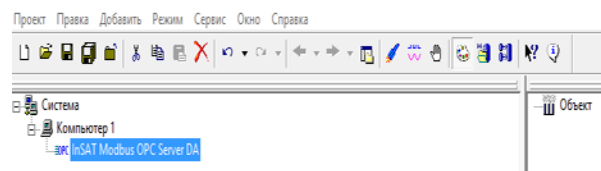


Рисунок 4. Создание связи между программами

Figure 4. The creation of linkages between the programmes

В) Щелкаем на «Insat ModbusOPCServerDA», появится окно выбора опций. В окне активируем опцию «Вставить OPC переменные». Выбрав эту опцию, получим окно свойств рисунок 5.

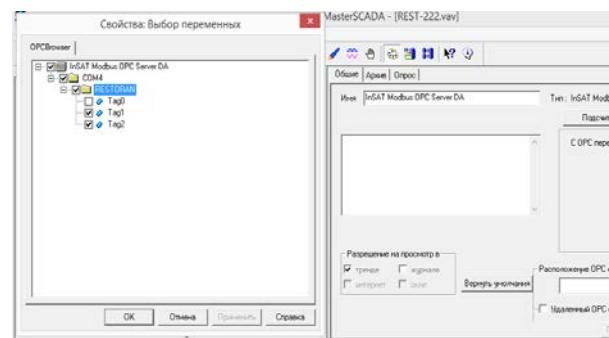


Рисунок 5. Окно для установки переменных из программы ардуино

Figure 5. The window to set the variables from the Arduino program

Г) В окне «Свойства: Выбор переменных» двойным щелчком мышки на кубике InSAtModbus OPCServer DA откроем подход к данным сервера и поочередно выведем все переменные, которыми мы оперировали, создавая OPC сервер (левое окно, рисунок 5). Закроем окно кнопкой «ОК».

В результате в окне панели системы «Scada», появятся наши рабочие датчики «tag1, tag2».

Далее, на любой мнемосхеме, мы сможем видеть показания наших датчиков температуры и объема жидкости, перенеся имена tag1, tag2 в окно выбранной схемы, которую мы создадим в MasterScada.

Пятый шаг – создаем мнемосхемы с объектами в скаде и проверяем работоспособность всего проекта.

Для создания профессиональных проектов в MasterScada, желательно использовать все возможности этой программы [4].

В данном случае описаны необходимые действия для получения мнемосхемы «Заставка» и второй мнемосхемы, где шеф повар наблюдает за работой пищеварочного котла.

Вначале создадим заставку, в которой есть меню выбора объектов наблюдения (котел, плита и т. п.). В качестве заставки удобно использовать фотографию Вашего горячего цеха.

Для заставки необходим файл в графическом формате bmp. Возможно, Вам потребуются в других мнемосхемах фотографии конкретных объектов (плит, шкафов), для использования в мнемосхемах годятся только файлы в указанном формате.

Для построения объектов переходим в окно дерева проектов, изменим название «объект» на имя ГОРЯЧИЙ и, щелкнув правой кнопкой мышки на этом имени, вставляем объект 1. Когда активизируем этот элемент (объект 1), правее дерева проектов появится новое меню, в котором есть опция «окна» (позиция 1 рисунок 6). Нажав кнопку «окна», выберем название мнемосхема и, нажав кнопку «Выбрать» (позиция 2), найдем наш графический файл горячего цеха. В позиции 3 установим галочку стартовой мнемосхемы.

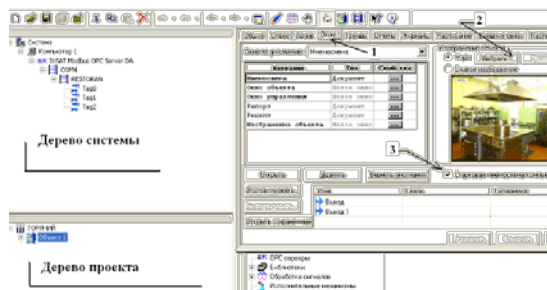


Рисунок 6. Этапы создания стартовой мнемосхемы
Figure 6. Steps to create a starter mimic

Далее подобным образом создаются следующие объекты программы. В нашем случае это схема котла, плиты и т. п. Каждая схема это новый объект в дереве проекта. Отличие лишь в том, что в следующих схемах мы должны видеть параметры работы установленных агрегатов. Напомним, что для передачи данных с датчиков в окно проекта, достаточно перенести имя (tag1, tag2) из дерева системы в окно созданного объекта. Пример для варочного котла на рисунке 7.

ЛИТЕРАТУРА

- 1 Русанов В.В. Экономические схемотехнические решения в системах автоматизации аграрного производства // Вестник РЭУ им. Г.В. Плеханова: 2016. № 6(90). С. 103–109.
- 2 Пехтин В.А. Проекты с использованием контроллера Arduino. СПб: БХВ-Петербург, 2015. 448 с.
- 3 ИнСАТ - Интеллектуальные Системы Автоматизации Технологии. URL: <https://www.Insat.ru/Modbus-OPC-Server>
- 4 MasterScada Основы проектирования. URL: https://www.masterscada.ru/files/art_step_by_step/Metod.pdf
- 5 Abdullah R. et al. Design an automatic temperature control system for smart tudungsaji using Arduino microcontroller // ARPN Journal of Engineering and Applied Sciences. 2016. V. 11. №. 16. P. 9578-9581.
- 6 Ya'acob N. et al. RFID lab management system using Arduino microcontroller approach associate with webpage // Journal of Scientific Research and Development. 2016. V. 3. №. 2. P. 92-97.

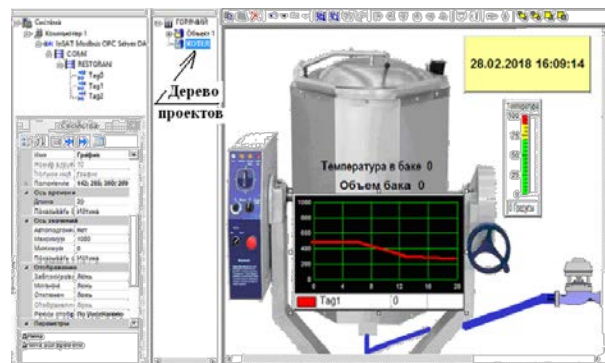


Рисунок 7. Вывод температуры, объема и времени варки, продукта в котле

Figure 7. Output of the temperature, volume and time of cooking, the product in the boiler

Причем, при выводе параметров есть возможность получения результатов в цифровой форме, в виде показаний стрелочных приборов, в виде графиков.

Последним действием создаем связь между двумя мнемосхемами. Для этого откроем окно мнемосхемы заставки (объект 1) и правой кнопкой мышки из дерева проектов перенесем туда второй объект (котел).

Заключение

Практическое воплощение всех представленных в статье действий позволит создать работоспособный проект. При запуске такой программы в режиме выполнения на экране появится заставка – фотография горячего цеха с кнопками перехода на другие объекты. Щелкнув на кнопке перехода на варочный котел, получим экран, представленный на рисунке 7, где возможно использование анимации объектов по событиям проходящего технологического процесса [4–10].

- 7 Al-Haija Q. A. et al. A tiny RSA cryptosystem based on Arduino microcontroller useful for small scale networks // Procedia Computer Science. 2014. V. 34. P. 639-646.
- 8 Erez N., Wool A. Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems // International Journal of Critical Infrastructure Protection. 2015. V. 10. P. 59-70.
- 9 Radvanovsky R., Brodsky J. et al. Handbook of SCADA/control systems security // CRC Press. 2016.
- 10 Huang S. C., Lu C. N., Lo Y. L. Evaluation of AMI and SCADA data synergy for distribution feeder modeling // IEEE Transactions on Smart Grid. 2015. V. 6. №. 4. P. 1639-1647.

REFERENCES

- 1 Rusanov V.V. Economical mnemotechnical decisions in the systems of automation agricultural production. Vestnik REU im. G.V. Plekhanova. [Proceedings of REU n.a. G.V. Plekhanov] 2016. no. 6(90). pp. 103–109. (in Russian)

2 Pekhtin V.A. Proekty s ispol'zovaniem kontrollera Arduino [Projects where Arduino is used. Saint-Petersburg. BHV-Petersburg] 2015. 448 p. (in Russian)

3 INSAT - Intelligent Systems of Automation Technologies. Available at: <https://www.Insat.ru/Modbus-OPC-Server> (in Russian)

4 MasterSSADA Design Basics. Available at: https://www.masterscada.ru/files/art_step_by_step/Metod.pdf (in Russian)

5 Abdullah, R., Rizman, Z. I., Dzulkefli, N. N. S. N., Ismail, S. I., Shafie, R., & Jusoh, M. H. (2016). Design an automatic temperature control system for smart tudungsaji using Arduino microcontroller. ARPN Journal of Engineering and Applied Sciences, 11(16), 9578-9581.

6 Ya'acob N., Yusof A. L., Azhar A. E., Naim N. F. et al. RFID lab management system using Arduino

microcontroller approach associate with webpage. Journal of Scientific Research and Development. 2016. no. 3(2). pp. 92-97.

7 Al-Haija, Q. A., Al Tarayrah M., Al-Qadeeb H., Al-Lwaimi A. A tiny RSA cryptosystem based on Arduino microcontroller useful for small scale networks. Procedia Computer Science. 2014. vol. 34. pp. 639-646.

8 Erez, N., Wool, A. Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems. International Journal of Critical Infrastructure Protection. 2015. vol. 10. pp. 59-70.

9 Radvanovsky R., Brodsky, J. Handbook of SCADA/control systems security. CRC Press, 2016.

10 Huang S. C., Lu C. N., Lo Y. L. Evaluation of AMI and SCADA data synergy for distribution feeder modeling. IEEE Transactions on Smart Grid, 2015. vol. 6. no. 4. pp 1639-1647.

СВЕДЕНИЯ ОБ АВТОРАХ

Владимир В. Русанов к.т.н., доцент, кафедра ресторанного бизнеса, Российский экономический университет им. Г.В. Плеханова, Стремянный пер.,36, г. Москва, 117997, Россия, kafedrates@mail.ru

Виталий И. Перов д.э.н., профессор, кафедра ресторанного бизнеса, Российский экономический университет им. Г.В. Плеханова, Стремянный пер.,36, г. Москва, 117997, Россия, perov-vitaliy@bk.ru

Михаил А. Самойлов бакалавр, ассистент, кафедра ресторанного бизнеса, Российский экономический университет им. Г.В. Плеханова, Стремянный пер.,36, г. Москва, 117997, Россия, Mi1995@yandex.ru

КРИТЕРИЙ АВТОРСТВА

Владимир В. Русанов предложил на базе трех программ автоматизировать работу горячего цеха, разработал кода для микроконтроллера

Виталий И. Перов проанализировал предлагаемое нововведение, записал программный код на микроконтроллер

Михаил А. Самойлов написал рукопись, корректировал её до подачи в редакцию и несёт ответственность за плагиат

КОНФЛИКТ ИНТЕРЕСОВ

Авторы заявляют об отсутствии конфликта интересов.

ПОСТУПИЛА 10.04.2018

ПРИНЯТА В ПЕЧАТЬ 07.05.2018

INFORMATION ABOUT AUTHORS

Vladimir V. Rusanov Cand. Sci. (Engin.), associate professor, restaurant business department, Plekhanov Russian University of Economics, Stremyanny per.,36, Moscow, 11997 Russia, kafedrates@mail.ru

Vitalii I. Perov Dr. Sci. (Econ.), associate professor, restaurant business department, Plekhanov Russian University of Economics, Stremyanny per.,36, Moscow, 11997 Russia, perov-vitaliy@bk.ru

Mikhail A. Samoilov bachelor, assistant, restaurant business department, Plekhanov Russian University of Economics, Stremyanny per.,36, Moscow, 11997 Russia, Mi1995@yandex.ru

CONTRIBUTION

Vladimir V. Rusanov suggested, on the basis of three programs, to automate the work of the hot shop, the development of code for the microcontroller

Vitalii I. Perov analyzed the proposed innovation, wrote the program code to the microcontroller

Mikhail A. Samoilov wrote the manuscript, correct it before filing in editing and is responsible for plagiarism

CONFLICT OF INTEREST

The authors declare no conflict of interest.

RECEIVED 4.10.2018

ACCEPTED 5.7.2018