

ИСПОЛЬЗОВАНИЕ LABVIEW ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ОСВЕТИТЕЛЬНОЙ УСТАНОВКИ

Беспалов Н. Н., Ильин М. В., Пьянзин И. И.

Мордовский государственный университет им. Н. П. Огарева,
Тел. 24-37-05. E-mail: bnn48@mail.ru, imike@rambler.ru, johnnie88@mail.ru.

Аннотация. В статье рассматривается программное обеспечение, предназначенное для управления осветительной установкой на основе светодиодных источников света, разработанное в среде программирования LabVIEW. В программном обеспечении имеется возможность задавать функцию освещения, изменяющуюся во времени, графически, а так же создавать, редактировать и просматривать файлы-сценарии освещения.

Ключевые слова. Осветительная установка, LabVIEW, виртуальный прибор, блок-диаграмма, файл-сценарий, цикл, редактор.

Summary. The article deals with software designed to control the lighting system based on LED light sources, developed in the programming environment LabVIEW. In software it is possible to specify the function of lighting, changing over time, graphically, as well as create, edit and view files, script coverage.

Keywords. Lighting system, LabVIEW, virtual device, block diagram, file script, loop editor.

Введение

В настоящее время перед инженерами все чаще встают задачи создания осветительных установок с высокой энергоэффективностью, которые, кроме этого, предоставляли бы пользователю максимальный комфорт и обладали простотой управления. Достижение этих целей трудно себе представить без применения в инженерной практике систем управления освещением. Программное обеспечение (ПО) систем управления освещением в основном предназначено на поддержание конкретного уровня освещенности в течение определенного промежутка времени, который, как правило, зависит от времени суток или времени года. Однако данные системы не позволяют реализовать динамически изменяющееся освещение, т.е. задать освещение какой-либо временной функцией.

Постановка задачи

Предполагается, что осветительная установка будет работать в цеху по выращиванию домашней птицы. В связи с этим, разрабатываемая программа должна обладать следующими возможностями и характеристиками:

- 1) автоматическое регулирование освещения в зависимости от времени;
- 2) создание, редактирование, просмотра сценария и его сохранение в файл;
- 3) простой и удобный в работе интерфейс.

Метод решения

Разработка программного обеспечения осветительной установки производилось в среде графического программирования LabVIEW. Данный язык программирования позволяет существенно снизить временные затраты на разработку ПО по сравнению с текстовыми языками программирования. Процесс разработки ПО в LabVIEW заключается в создании программного кода, который в достаточной степени похож на алгоритм. Такое представление кода, разрабатываемой программы, имеет высокий уровень наглядности, что существенно позволяет упростить процесс разработки ПО. Программа, разработанная в LabVIEW, называется виртуальный прибор (ВП). ВП состоит из двух частей. Первая часть

является непосредственно программным кодом, а вторая – лицевая панель, которая выполняет функции интерфейса программы. Основное требование при разработке лицевой панели заключается в создании максимально информативного и удобного при работе интерфейса ПО, который будет являться основным, разрабатываемого оборудования.

Алгоритм работы программы представляет собой циклическое повторение действий предназначенных для определения текущего момента времени и соответствующего значения освещенности в данный момент времени на основе ранее заданного сценария. Сценарий представляет собой набор временных зависимостей освещенности, описывающих каждый цикл регулирования освещения.

Программа выполнена по шаблону «Событийно управляемый конечный автомат с параллельными циклами». Структура программы в LabVIEW показана на рисунке 1.

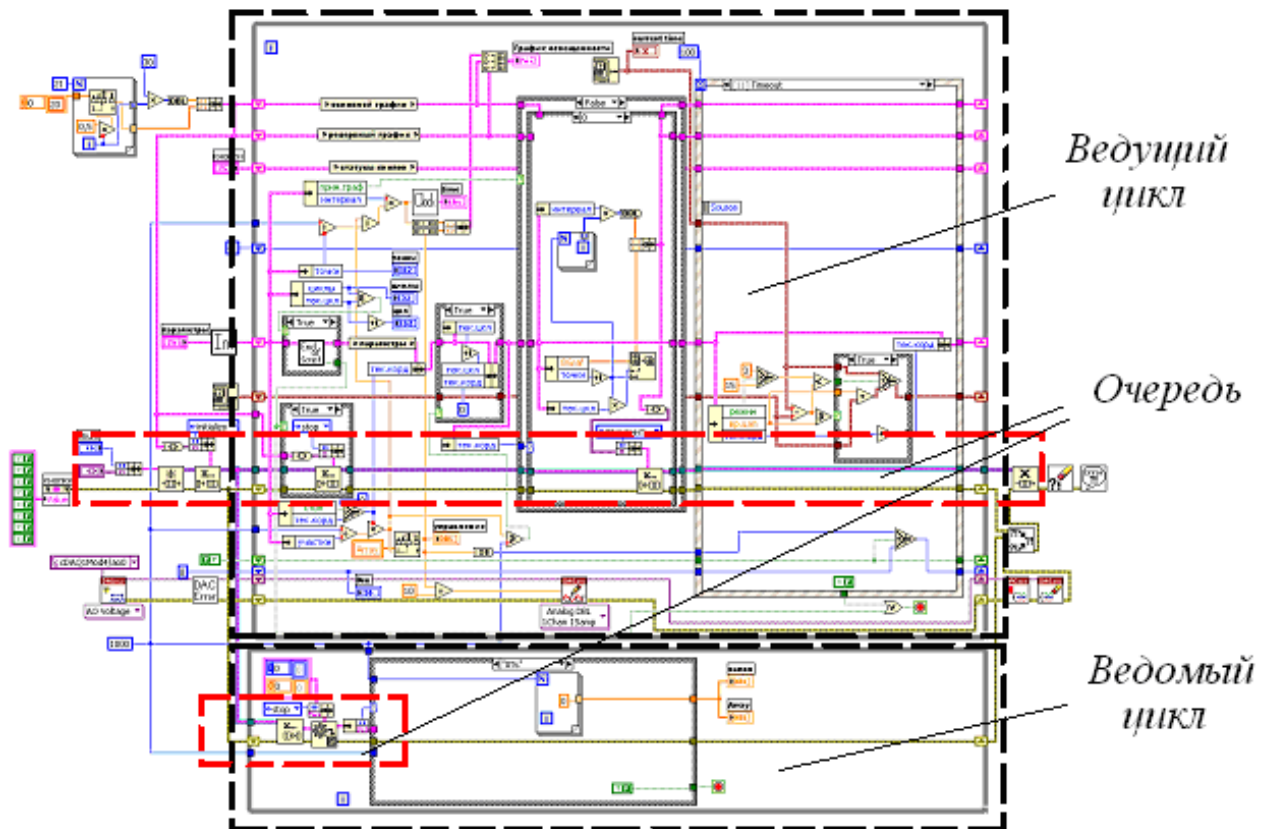


Рис. 1. Структура программы в LabVIEW.

Верхний цикл While Loop по отношению к нижнему является ведущим. В LabVIEW ведомый цикл не может выполняться параллельно ведущему, пока тот не завершится, поэтому, для его запуска, используется очередь «Queue». Она принудительно запускает ведомый цикл и передаёт нужные данные из верхнего в нижний цикл. Данные из нижнего цикла передаются в верхний через локальную переменную. Ведущий цикл в данной программе выполняет несколько следующих функций:

- 1) системный таймер;
- 2) расчёт уровня освещённости в текущий момент и преобразование его в величину управляющего сигнала;
- 3) вывод величины управляющего сигнала в цифро-аналоговый преобразователь (ЦАП);
- 4) обработка событий интерфейса.

Первая функция реализует таймер, который является объектом временной привязки для всех процессов в программе. По значениям таймера рассчитывается величина освещённости на основе установленной временной функции в настройках. Рассчитанная величина преобразуется в соответствии с требуемым законом в величину управляющего

сигнала, которая выводится в ЦАП. В данном случае величина освещённости перерасчитывалась на основе линейного закона. В качестве ЦАП может быть применено любое соответствующее оборудование компании «National Instruments». При разработке данного устройства применялся модуль С-серии NI 9263 совместно с устройством cDAQ. События интерфейса обрабатываются структурой Event. В данном случае под событиями интерфейса подразумевается обработка кнопок на лицевой панели.

Лицевая панель ВП содержит индикаторы, отображающие основные параметры состояния системы управления и кнопки позволяющие управлять данным ВП. Лицевая панель виртуального прибора представлена на рис. 2.

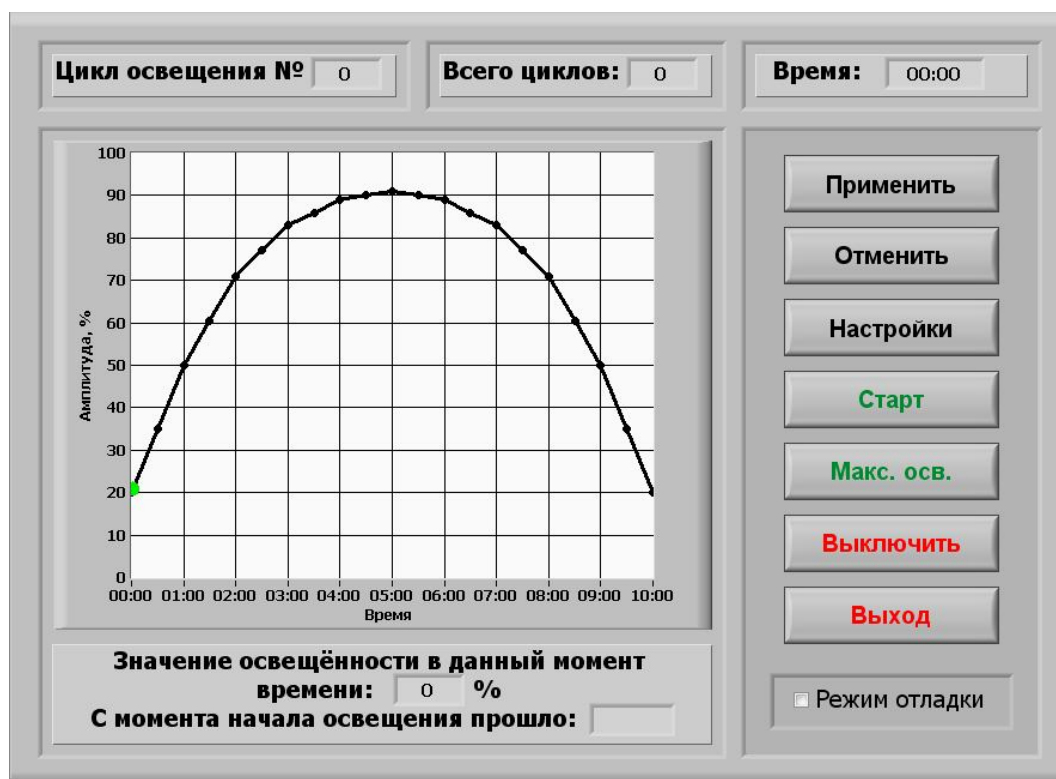


Рис. 2. Лицевая панель ВП.

В верхней части лицевой панели отображаются текущие параметры сценария процесса регулирования освещения, к которым относятся номер цикла освещения, количество циклов, и текущее время цикла. Текущая временная зависимость отображается в графическом виде на лицевой панели. Дополнительно отображается и текущий момент на графике в виде зелёной точки, которая перемещается по графику в соответствии с текущим моментом времени.

Текущую временную зависимость возможно изменять непосредственно на лицевой панели с помощью мышки перемещая опорные временные точки на необходимые уровни освещённости. Для утверждения или отмены изменений на лицевой панели имеются кнопки «Применить» и «Отменить». По нажатию кнопки «Применить» изменения графика применяются к графику освещения. Кнопка «Отменить» наоборот, отменяет изменения и возвращает график к предыдущему виду.

При нажатии кнопки «Настройки» запускается подприбор позволяющий редактировать параметры сценария освещения. На рис. 3. представлена лицевая панель данного подприбора. В данном подприборе можно задать такие параметры сценария освещения, как количество циклов освещения сценария, их протяженность, а так же указать, с какого цикла и времени нужно начать процесс освещения. Помимо этого здесь можно создать, отредактировать, просмотреть и загрузить файл-сценарий в основную программу.

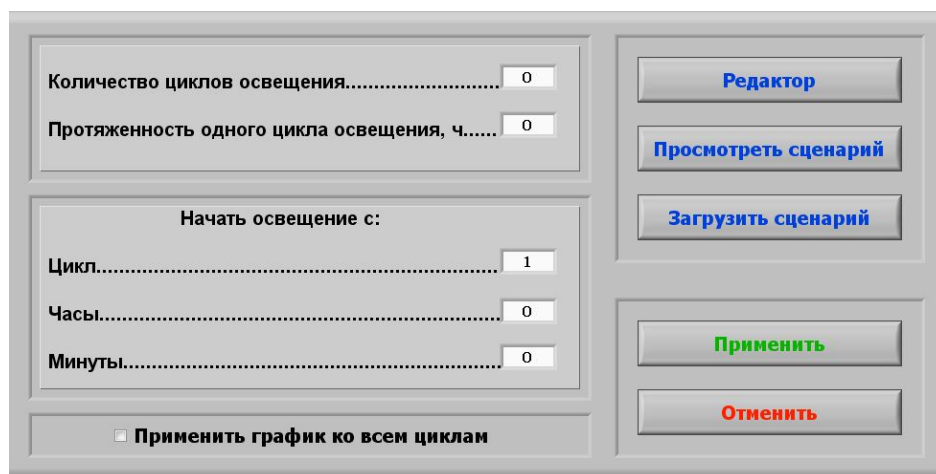


Рис. 3. Лицевая панель подприбора редактора сценария.

Кнопка «Старт» на основном ВП запускает систему управления осветительной установки в соответствии с установленным сценарии освещенности. По кнопке «Макс.осв.» на ЦАП подается значение, эквивалентное освещенности в 100%. По кнопке «Выключить» - значение, эквивалентное 0% освещенности. Кнопка «Выход» завершает программу.

Редактор файла сценария

Файл-сценарий представляет себя файл «двоичного кода» (Binary File), в котором содержится кластер из 6 элементов. Первые 4 элемента, это переменные «интервал», «точки», «циклы», «часы». Эти переменные хранят целочисленные значения в формате I32 (integer 32bit). Остальные 2 элемента – это массивы. Первый массив состоит из чисел формата DBL (Double Precision – двойная точность), второй состоит из логических констант. Массив чисел (массив графиков) хранит в себе ординаты графиков всех циклов. Массив логических элементов (массив заполненности) хранит информацию о том, какие циклы были заполнены, а какие – нет. На рис. 4 показана структура кластера файла-сценария.

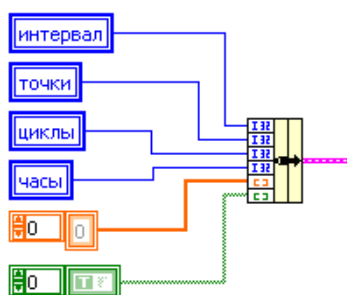


Рис. 4. Структура файла-сценария.

Создать файл-сценарий можно с помощью подприбора редактора сценария, доступ к которому осуществляется через «Настройки» → «Редактор» → «Создать сценарий». После выбора количества циклов и их протяженности откроется лицевая панель подприбора редактора файла сценария, показанная на рис. 5. Редактирования текущего графика сценария так же осуществляется с помощью мышки, перемещая опорные временные точки.

Кнопки «Применить» и «Отменить», по аналогии с одноименными кнопками в основной программе, выполняют функции применения или отмены изменений вида графика.

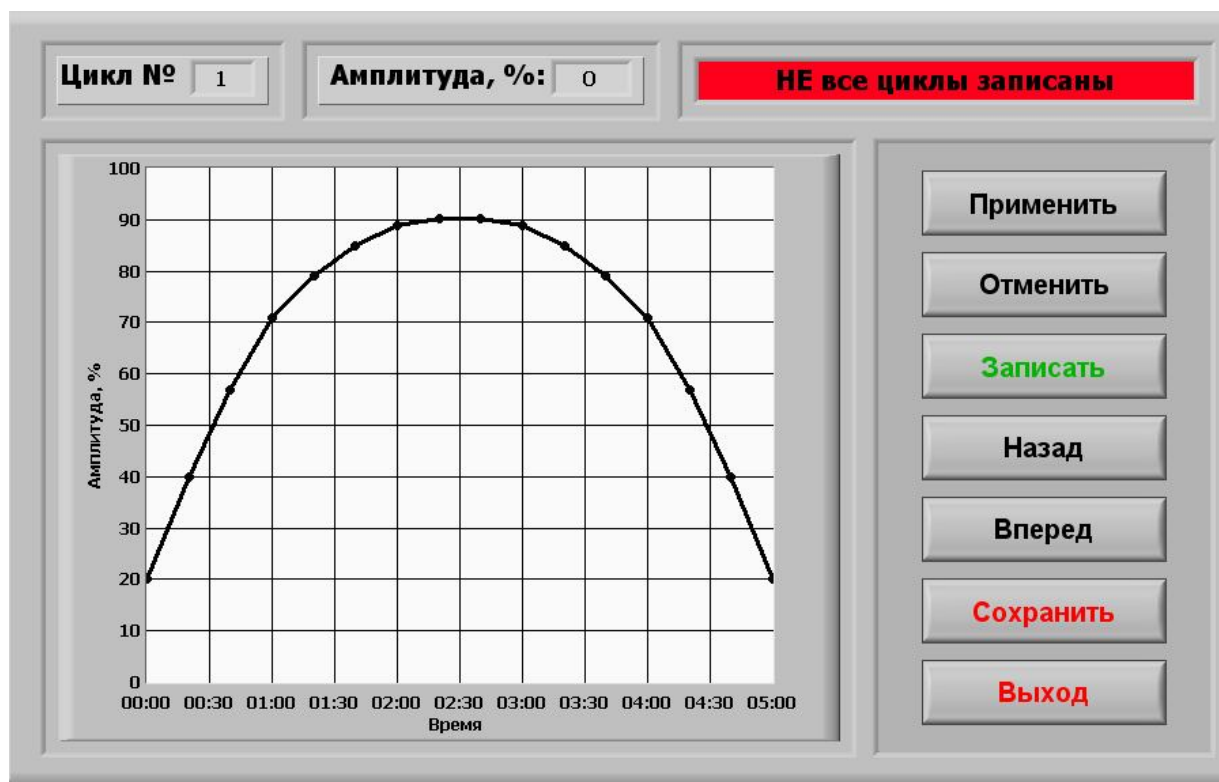


Рис. 5. Вид редактора сценария.

По кнопке «Записать» координаты графика передаются в массив чисел, и происходит переход к редактированию графика следующего цикла. Помимо этого, вид графика записанного цикла переносится на следующий. Элементу массива заполненности по номеру записанного цикла присваивается значение True. Если все циклы будут записаны, то сообщение в верхнем правом углу сменится на «Все циклы записаны».

Кнопки «Назад» и «Вперед» осуществляют переход на предыдущий и следующий циклы.

При нажатии на «Сохранить» открывается окно, где пользователю нужно будет выбрать путь к файлу, в который будет записан сценарий. После выбора директории данные записываются в этот файл.

При нажатии на кнопку «Выход» редактор прекращает свою работу. Если массив графиков был изменен с момента последнего сохранения, то после нажатия на эту кнопку открывается сообщение, где пользователю предлагается сохранить измененные данные или выйти без сохранения.

Редактирование готового сценария осуществляется с помощью дополнительного виртуального подприбора. Он запускается из окна настроек путём последовательного нажатия на кнопки «Редактор» и «Редактировать готовый сценарий». После этого запускается окно, в котором пользователю нужно выбрать файл, сценарий которого он хочет изменить. Далее запускается редактор, который отличается от предыдущего лишь только тем, что, после нажатия на кнопку «Записать», записанный график не передаётся на последующий цикл. Лицевая панель этого редактора абсолютно идентична панели предыдущего.

Просмотреть файл сценария можно с помощью подприбора предназначенного для просмотра сценариев. Для его запуска нужно нажать кнопку «Просмотреть сценарий» на панели настроек. После этого откроется передняя панель этого прибора. Она представлена на рис. 6.

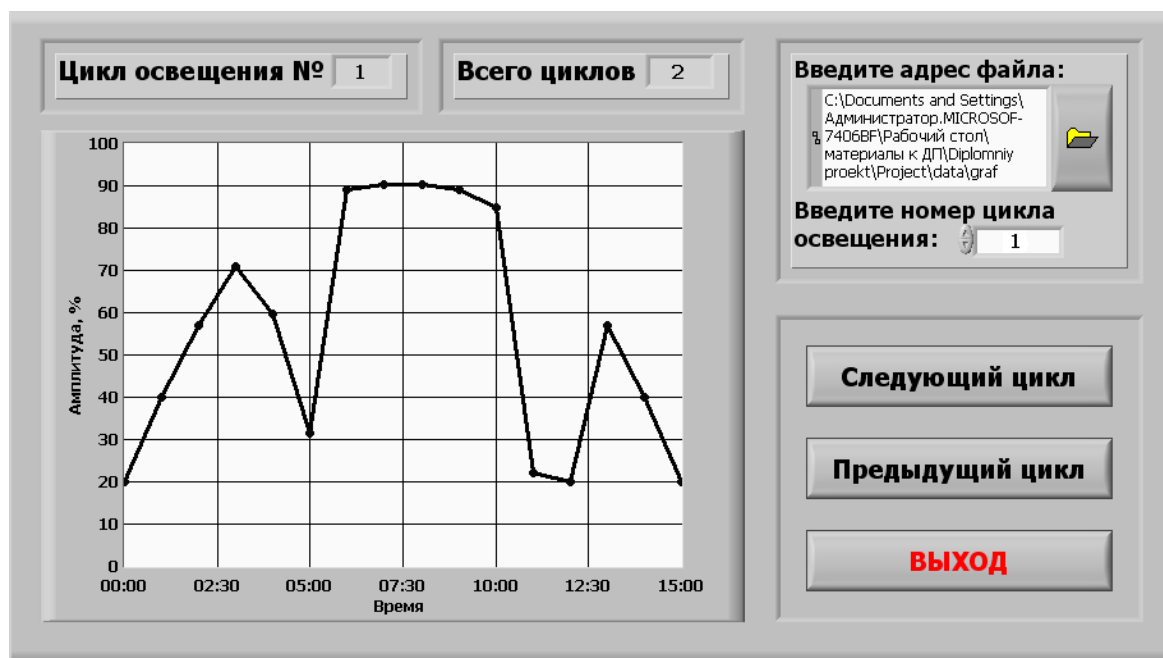


Рис. 6. Окно ВП «prop_viewer.vi».

В правом верхнем углу панели находится терминал выбора файла сценария, а так же терминал номера цикла, который можно ввести с клавиатуры. Помимо этого терминала, просмотреть графики других циклов сценария можно с помощью кнопок «Следующий цикл» и «Предыдущий цикл». После нажатия кнопки «Выход» данный ВП прекращает свою работу.

Заключение

В ходе работы был разработан виртуальный прибор, с помощью которого реализуется управление осветительной установкой в птицеводческом цехе. Среда визуального программирования LabVIEW позволила создать приложение с удобным и понятным пользователю интерфейсом.

Список используемой литературы

1. Блюм П., LabVIEW: стиль программирования. Пер. с англ. под ред. Михеева П. – М.: ДМК Пресс, 2008 – 400 с.: ил.;
2. Тревис Дж., LabVIEW для всех / Джеффри Тревис : Пер. с англ. Клушин Н. А. – М.: ДМК Пресс; ПриборКомплект, 2004. – 544 с.: ил.

Сведения об авторах

Беспалов Николай Николаевич – кандидат технических наук, заведующий кафедрой автоматике Мордовского государственного университета им. Н. П. Огарёва, e-mail: bnn48@mail.ru.

Ильин Михаил Владимирович – кандидат технических наук, доцент кафедры автоматике Мордовского государственного университета им. Н. П. Огарёва, e-mail: imike@rambler.ru.

Пьянзин Иван Иванович – студент Мордовского государственного университета им. Н. П. Огарёва, e-mail: johnnie88@mail.ru.