

## РАЗРАБОТКА СУПЕРВИЗОРА СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ LabVIEW

**Воронова А.И., студент; Мальчева Р.В., доц., к.т.н.; Авксентьева О.А., асс.**  
(ГВУЗ «Донецкий национальный технический университет», г. Донецк)

**LabVIEW** (англ. **L**aboratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench) – это среда разработки и платформа для выполнения программ, созданных на графическом языке программирования G фирмы National Instruments (США). Первая версия LabVIEW была выпущена в 1986 году для Apple Macintosh, в настоящее время существуют версии для UNIX, Linux, Mac OS и пр., а наиболее развитыми и популярными являются версии для Microsoft Windows. Исходные тексты переносимы, а программы будут выглядеть одинаково во всех системах. Код, сгенерированный LabVIEW, также может быть исполнен на Windows Mobile или PalmOS (на данный момент поддержка PalmOS прекращена). Этот язык может с успехом использоваться для создания больших систем, для обработки текстов, изображений и работы с базами данных.

Существует два основных отличия LabVIEW от других сред. Во-первых, LabVIEW реализует концепцию графического программирования G, поэтому исходный код представляет собой блок-диаграмму (соединённые друг с другом пиктограммы элементов языка), которая затем компилируется в машинный код. Несмотря на такой подход в языке G используются те же конструкции и методы программирования, что и в других языках: типы данных, циклы, переменные, рекурсия, обработка событий и объектно-ориентированное программирование.

Вторая отличительная особенность LabVIEW – это поддержка выполнения кода, написанного на языке G, в режиме потока данных (потокосное программирование), в то время как традиционные текстовые языки (например, C и C++) обеспечивают выполнение кода в виде последовательности команд. В основе языков потокосного программирования (таких как G, Agilent VEE, Microsoft Visual Programming Language и Apple Quartz Composer) лежит концепция потока данных, который и определяет последовательность выполнения функциональных узлов программы.

В данной работе рассматривается разработка среды тестирования и отладки работы супервизора системы реального времени и непосредственно супервизора задач реального времени. Программа создана на языке G, следовательно, выполняется в режиме потока данных. Однако часто необходимо, чтобы некоторые действия выполнялись в строгой последовательности. В таких случаях используют структуры последовательных действий (Flat Sequence, Stacked Sequence). При использовании таких структур, помещенные в них операторы выполняются «покадрово».

На рис. 1 приведена лицевая панель разрабатываемой LabVIEW-программы. Лицевая панель включает несколько функциональных групп элементов управления и отображения:

- группа управляющих клавиш («Пуск», «Пауза», «СТОП», «Аварийный сброс»), отвечающих за режимы работы разрабатываемого ВП;
- группа задания параметров задач системы, состоящая из четырёх кластеров управляющих элементов;
- группа индикаторов системного времени в составе индикатора времени моделирования и индикатора времени с начала работы, расширенная численным полем ввода – множителем частоты работы ВП «Ускоритель»;
- область вывода текущего состояния задач в системе с индикаторами активности задач;
- область вывода расчётного значения;
- строка ввода и отображения нажатой клавиши.

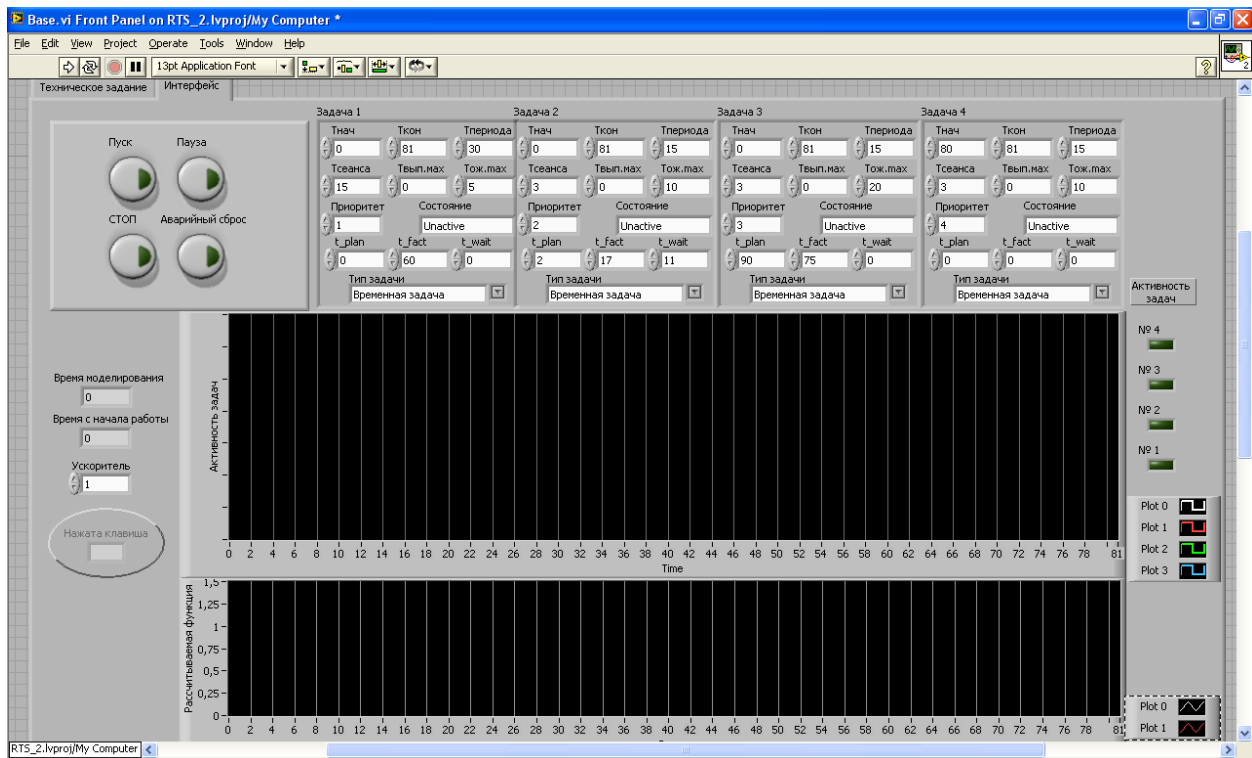


Рисунок 1 – Лицевая панель программы тестирования и отладки супервизора задач реального времени

Для организации управления ходом работы ВП введено 4 клавиши, выполняющие следующие функции:

Клавиша «Пуск» – клавиша типа «Switch when pressed», переводящая ВП в активное или неактивное состояния в соответствии с положением клавиши в нажатом или отжатом положениях. При нажатой клавише счётчик времени с начала работы активен, поля ввода параметров задач не подлежат изменению, а фокус ввода передаётся строке ввода символов. Счётчик времени моделирования запущен, а супервизор, анализирующий его значение, управляет задачами в соответствии с их параметрами. Область вывода состояния задач отображает их активность.

Клавиша «Пауза» – клавиша типа «Switch when pressed», приостанавливающая процесс моделирования. Счётчик времени с начала работы ВП не останавливается, поле строки ввода символов неактивно.

Клавиша «Аварийный сброс» – клавиша типа «Switch until released», по нажатию – сброс времени моделирования в 0, перевод клавиши «Пауза» в нажатое состояние, очистка области вывода состояния задач, счётчик времени с момента запуска не останавливается и не обнуляется, при ненажатой клавише «Пуск» открывается возможность изменения параметров задач.

Клавиша «СТОП» – клавиша типа «Switch until released», по нажатию – прекращение выполнения LabVIEW-программы.

*Группа задания параметров задач системы:* параметры каждой задачи объединены в отдельный кластер – структуру, позволяющую задавать свойства всем включённым в неё параметрам, а так же вносящей в определённом смысле «порядок» в блок-диаграмму, повышая читаемость алгоритма. Параметры задач в рассматриваемом случае не изменяются во время моделирования, переводясь в состояние «Disabled» при времени моделирования неравном нулю, защищающее поля кластеров от ввода, и не мешающее, тем не менее, чтению текущих параметров задач.

*Группа контроля системного времени:* цифровые индикаторы отображают значения переменных времени моделирования и времени с начала работы ВП. Поскольку время моделирования может быть достаточно длительным, введён числовой управляющий элемент

«Ускоритель», с помощью которого временная задержка в каждой итерации цикла ВП может быть увеличена для более удобного анализа получаемых результатов в процессе моделирования или уменьшена для более быстрого выполнения моделирования.

*Область вывода текущего состояния задач в системе:* объект «Waveform Chart» выводит состояние каждой задачи в каждый системный момент времени, индикаторы активности введены для лучшего восприятия информации о состоянии задачи, а также для контроля корректности выводимых результатов. При запуске ВП и сбросе история выводимых данных очищается.

*Область вывода расчётного значения:* объект «Waveform Chart», который предназначен для вывода результата работы некоторой активной задачи. При запуске ВП и сбросе очищается.

*Строка ввода и отображения нажатой клавиши* представляет собой строковый управляющий элемент, который предназначен для приёма символа от клавиатуры с целью последующего анализа и запуска соответствующей задачи. Получает фокус в процессе моделирования, т.о., символы всех нажатых клавиш заносятся в строку.

Структурно программа состоит из двух основных кадров: кадра начальной инициализации данных и кадра непосредственно рабочего алгоритма программы. «Рабочий» кадр организован на основе цикла «While», выход из которого происходит по нажатию кнопки «СТОП». В первом кадре последовательно выполняется начальная инициализация полей данных значениями «по умолчанию», и лишь затем выполняется последующая часть алгоритма. Добавление структур последовательных действий облегчает отладку программы, поскольку по умолчанию операторы выполняются по мере поступления данных, а в случае их наличия сразу на входах нескольких операторов – в случайном порядке. Ввиду этого в основном ВП в блок-диаграмму добавлены структуры Flat Sequence, отсутствие которых не повлияло бы на правильность выполнения алгоритма.

В каждой итерации происходит чтение полей входных данных задач и анализ корректности их содержимого. Анализ выполняется специальным виртуальным подприбором «check\_input.vi» (рис. 2), результатом работы которого являются корректные исходные данные, которые перезаписываются в поля ввода параметров задач. В данном виртуальном подприборе анализируются данные в контрольных полях и, при введённых в поля управления параметрами задач неверных значениях, генерируется кодовый номер типа и подтипа ошибки, запускается подприбор-анализатор ошибок «Error\_analyzer.vi».

Подприбор «Error\_analyzer.vi» инициализирует вызов диалогового окна с сообщением о типе ошибки и номер задачи, в параметрах которой имеются некорректные поля. Подприбор возвращает значение «true» по нажатию пользователем кнопки «ОК», свидетельствующем об его осведомлении о факте допущенной ошибки в заполнении параметров задачи, по которому ВП «check\_input.vi» корректирует параметры задач и подаёт на выходные терминалы корректные кластера параметров задачи.

Значения модельного времени и времени с начала работы основного ВП формируются в ВП «Timer.vi» (рис. 3).

Срабатывая, как и супервизор, на порядок чаще, чем остальные блоки программы, таймер позволяет сформировать значение текущего модельного времени с необходимой супервизору точностью.

Значения модельного времени и времени с момента начала работы заносятся в глобальные переменные и доступны для чтения в пределах всех виртуальных подприборов основного ВП и непосредственно в нём. Изменения в эти переменные вносятся только таймером и обработчиком нажатия кнопки «Аварийный сброс», а при начале работы программы переменные инициализируются нулями.

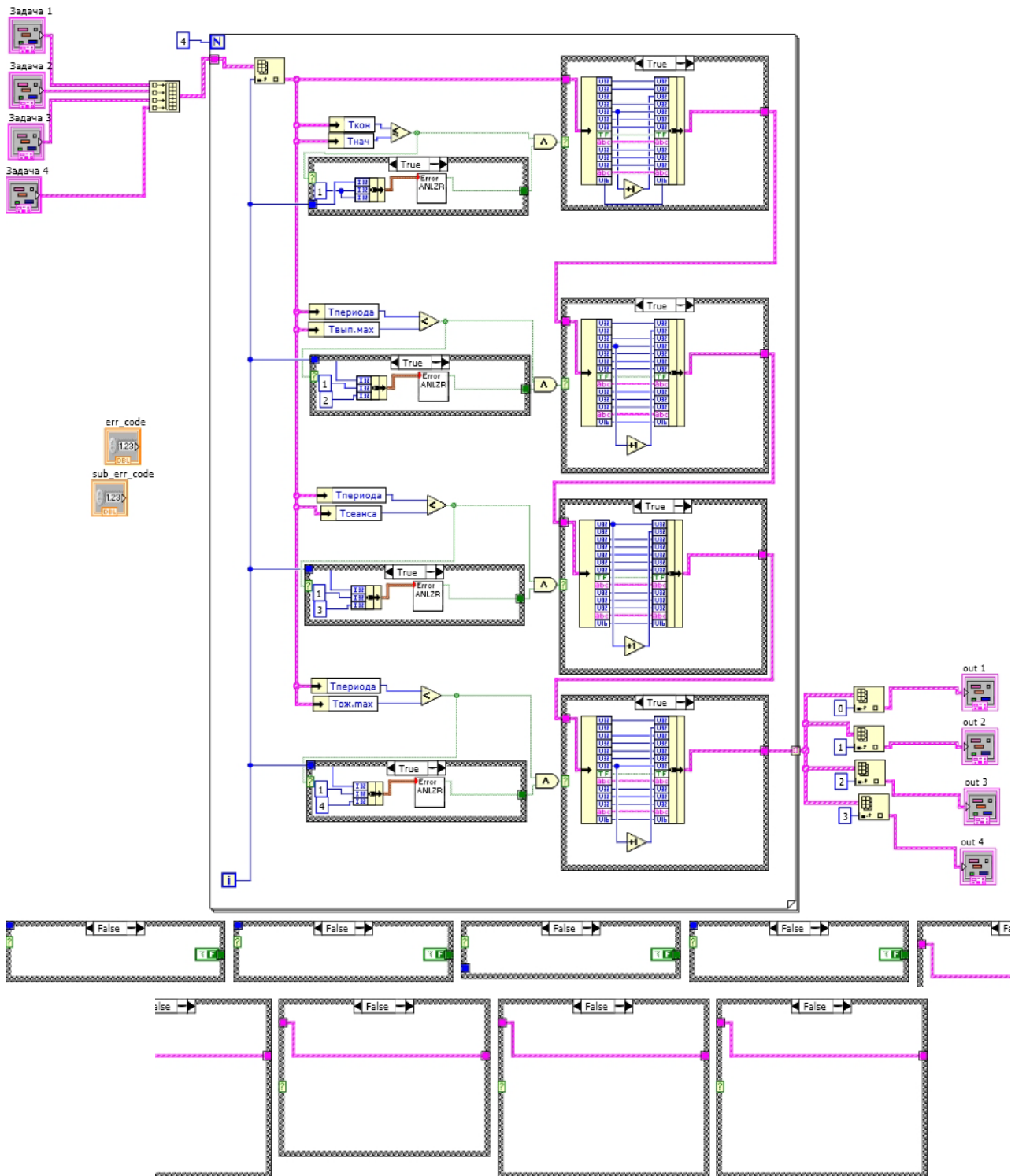


Рисунок 2 – Блок-диаграмма виртуального прибора check\_input.vi

Корректные параметры задач поступают в супервизор – ВП «Supervisor2+.vi», алгоритм которого на основании текущего модельного времени формирует в массиве данных признаки разрешения выполнения задачи, перевода её в состояние ожидания, приостановление или же прекращение работы, анализирует превышение времени выполнения задачи, времени ожидания запуска, формирует для каждой задачи плановое время вызова и фиксирует фактическое время вызова задачи. Результаты работы супервизора заносятся в переменную «TaskData», а по окончании работы супервизора заносятся в соответствующие поля задач.

Параллельно с основным циклом выполняется цикл вывода состояния задач, моделирующий их реальную работу. Таким образом, используется принцип многопоточности языка «G».

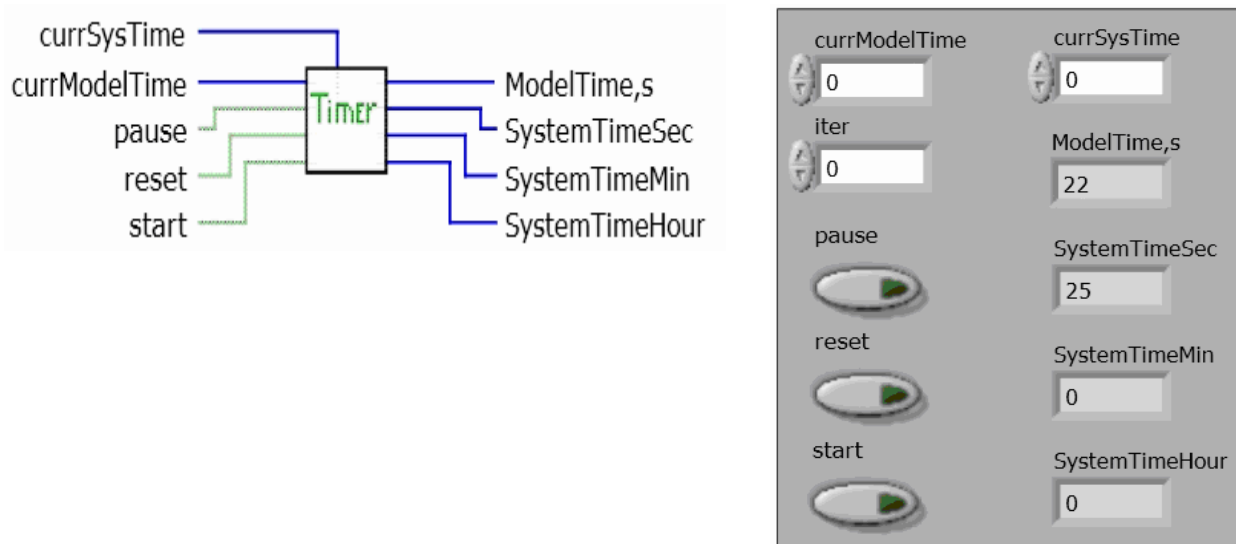


Рисунок 3 – Иконка и лицевая панель подбиратор-анализатор ошибок «Error\_analyzer.vi»

Супервизор или диспетчер задач в системе реального времени является неотъемлемой составляющей. Его задача состоит в анализе информационной базы параметров реального времени и поступающих в систему внешних сигналов, которые должны быть обработаны задачами-обработчиками данных сигналов. По результатам оценки супервизор оценивает необходимость изменения состояния прикладных задач и заносит вычисленные параметры в информационную базу, в которой устанавливается код состояния для требуемой задачи-обработчика.

Оценка полей информационной базы реального времени и изменение состояния задач происходит после алгоритмической обработки информационной базы супервизором, т.е. сам супервизор непосредственно не вносит изменений в состояние задач, а лишь взводит условные флаги изменения параметров задач.

Часто при разработке ВП использование лишь графического представления нецелесообразно. В случае с рассматриваемым ВП реализация алгоритма анализа полей данных задач с помощью лишь стандартных графических LabVIEW-операторов не целесообразна, такой вариант был бы слишком громоздким и трудночитаемым. Выходом из данной ситуации служит использование модулей кода других языков программирования.

При разработке данного ВП были использованы как средства языка «G», так и модуль C-кода, помещённый в элемент «Formula Node».

Результатом выполнения данной работы является LabVIEW-программа, моделирующая в общих чертах работу супервизора в системе реального времени. Данная работа стала полигоном для исследования специфики работы в среде разработки LabVIEW. В ходе её выполнения были рассмотрены базовые операторы графического языка «G», рассмотрены возможности комбинирования графического потокового и текстового императивного языков для реализации единой задачи, особенности построения и выполнения потоковой программы. Практически установлено, что специфические задачи, сродни поставленной в данной работе, целесообразно решать с использованием рассматриваемой платформы.

#### Перечень ссылок

1. <http://russia.ni.com/labview>
2. <http://www.labviewportal.eu>