

## Уроки по LabVIEW

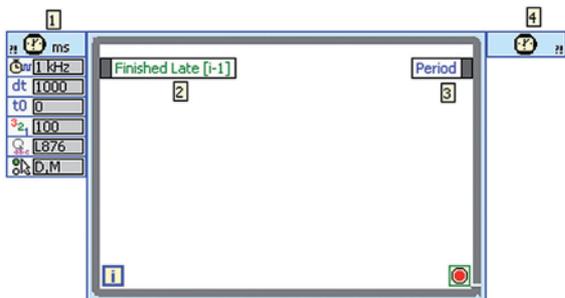
"While Loop", "For Loop", условные штуки типа "Case Structure" и многие другие, рассмотренные в предыдущих уроках, – все это для начинающих "программистов". А Вы, будем надеяться, уже освоили базовый курс LabVIEW в полном объеме. Давайте-ка на этом уроке рассмотрим новый операнд "Timed Loop", который позволяет оптимизировать исполняемый код. Структуру Timed Loop можно считать новинкой. Впервые она появилась в версии LabVIEW 7.0. Все необходимые пиктограммы находятся в палитре "Structure >> Timed Loop".



### Синхронизированные циклы

Итак, Вы знаете, что для выполнения циклического процесса до тех пор, пока не будет выполнено условие выхода из цикла, используется оператор "While Loop". В отличие от "While Loop", в "Timed Loop" предусмотрена возможность определения времени для каждой итерации, которая выполняется в ходе выполнения цикла. Если Вы хотите создать программу с переменной скоростью выполнения, точным временем выполнения, индикацией времени выполнения каждой итерации цикла, динамически изменяемым временем выполнения или несколькими уровнями выполнения, без "Timed Loop" не обойтись.

Рассмотрим основные элементы, которые Вы можете увидеть сразу после размещения этой структуры на диаграмме. Она содержит четыре компонента: 1 - входной терминал, 2 - левый терминал данных, 3 - правый терминал данных и выходной терминал данных - 4.



Входной терминал предназначен для конфигурирования "Timed Loop". Значения "по умолчанию", естественно, хранятся в "Configure Time Loop", но об этом чуть позже.

Левый терминал данных предназначен для задания времени и получения информации о выполнении предыдущего цикла, такой как запаздывание во время исполнения, точное время старта цикла и время его завершения. Вы можете соединить левый терминал данных с правым терминалом данных для задания конфигурации следующего цикла. Щелкните правой клавишей манипулятора на границе "Timed Loop" и выберите из выпадающего меню пункт "Show Right Data Node" для отображения правого терминала данных. Можно также изменить размеры левого и правого терминалов данных для получения дополнительных возможностей.

Выходной терминал данных возвращает код ошибки, получаемой "Timed Loop" из входного терминала, или код ошибки, возникающей в "Timed Loop" во время выполнения. Например, если внешний источник данных прерывает работу "Timed Loop", или если работают более 128

"Timed Loop" одновременно, в выходном терминале появляется код ошибки. Выходной терминал не возвращает код ошибки подпрограммы, работающей совместно с "Timed Loop".

### Конфигурирование

Щелкните два раза левой клавишей манипулятора по входному терминалу, или щелкните правой клавишей манипулятора по входному терминалу и выберите из выпадающего меню "Configure Timed Loop". Теперь можете конфигурировать свой "Timed Loop" и задать условия выполнения цикла. Это диалоговое окно служит для указания времени синхронизации источника, периода исполнения, смещения времени старта и других опций. Для использования внешних источников данных сделайте обязательно отметку в пункте "Use terminal" - подсоединить управляющие элементы можно только к терминалам, для которых активна опция "Use Terminal". Любые опции, которые Вы задаете в диалоговом окне "Loop Configuration" будут отображены на иконке входного терминала при условии правильного задания вида терминала. А для этого следует щелкнуть правой клавишей манипулятора на входном терминале, выбрать пункт "Name Format", а в нем - опцию "Icons".

И еще о конфигурировании. Правый терминал данных можно использовать для динамического изменения периода, смещения старта по времени, задания приоритета. Обновления вступают в силу на следующем шаге цикла.

### Синхронизация

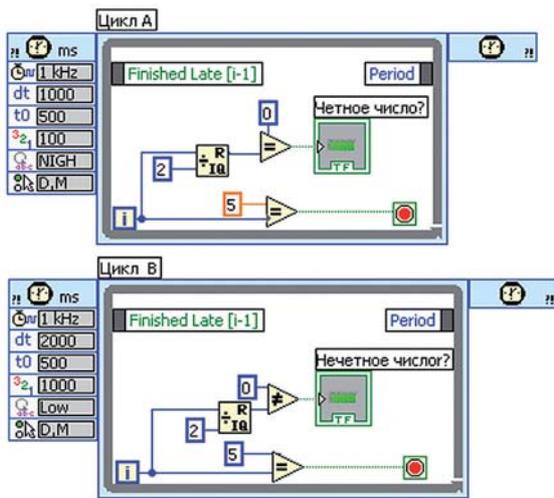
Источник, задающий время исполнения синхронизированных циклов "Timed Loop", это системный таймер, используемый операционной системой, формирующий минимальный интервал в одну миллисекунду. Если же система содержит дополнительные средства контроля времени (LabVIEW Real Time), то у Вас появится возможность задать такт длительностью 1 мкс! Выбор источника синхронизации находится в меню "Loop Configuration" - "Source type"

### Установка периода и временного сдвига

Параметры, определяемые блоком "Timing source", используются для задания периода исполнения цикла "Period" и временного сдвига "Offset", т.е. времени ожидания начала цикла.

На приведенной ниже блок-диаграмме циклы "Timed Loop" используют источник синхронизации часто-

той 1 ГГц и ожидание старта "t0", равное 500. Так образом, циклы будут находиться в состоянии ожидания 500 мс, прежде чем начнется их выполнение.



Период "dt" для цикла А равен 1000 мс, а период для цикла В - 2000 мс. Это означает, что цикл А будет выполняться один раз в секунду, а цикл В - один раз в две секунды. Число итераций в обоих циклах равно 6. Поэтому останов цикла А произойдет через 6 с, а цикла В - через 12 с.

### Установка приоритета

Каждый цикл "Timed Loop", показанный на блок-диаграмме, создается и выполняется в своем собственном потоке "Thread". Приоритет "Priority" цикла "Timed Loop" указывает, который из циклов выполняется первым по отношению к другим. Если Ваш VI содержит несколько циклов "Timed Loop", которые выполняются каждый в своем потоке, то Вы получаете реально многозадачное приложение. Используйте для этого установку приоритета цикла "Timed Loop".

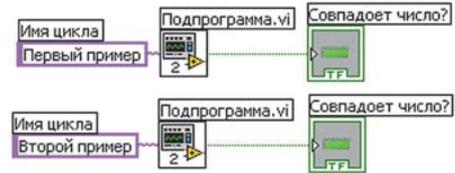
Оба цикла А и В, в рассмотренном выше примере имеют одинаковую задержку старта "t0" и одинаковые приоритеты. В этой ситуации второй цикл запускается после того, как первый стартовал. Но если Вы установите значение приоритета цикла В равным 1000, а этот приоритет больше чем у цикла А, то цикл В будет выполняться первым.

Следует заметить, что цикл "Timed Loop" имеет более низкий приоритет, чем приоритет "Time-critical", задаваемый для VI. Т.е. сначала выполняется VI, для которого задан приоритет "Time-critical", а потом выполняется "Timed Loop", для которого задан наибольший приоритет. Но если программа не имеет приоритета "Time-critical", то "Timed Loop" выполняется в потоке блок-диаграммы первым.

### Присвоение имени

По умолчанию циклу "Timed Loop" автоматически присваивается индивидуальное имя. Идентифицируется каждый цикл, который находится на диаграмме. Но это имя можно изменить, используя диалоговое окно "Timed Loop Configuration". Причем Вы можете использовать индивидуальное имя для программной остановки цикла и для синхронизации группы циклов.

Если запускаемый VI содержит цикл "Timed Loop" и на блок-диаграмме таких VI используется два и более, то Вы обязательно должны изменить имена "Timed Loop", содержащиеся в них. Не забывайте - нельзя использовать несколько циклов с одинаковыми именами! В этом случае для цикла "Timed Loop" необходимо вывести терминал индивидуального имени и изменить имя. Если Вы создали подпрограмму, в которой находится "Timed Loop", то и в этом случае каждому циклу должно быть присвоено индивидуальное имя. Если этого не сделать, то естественно, возникнет ошибка исполнения программы.



### Режимы работы цикла Timed Loop

Иногда цикл "Timed Loop" может выполняться дольше, чем задано. Откорректировать такую ситуацию можно, используя опцию из секции "Timed Loop action to take on late iterations" в конфигурационном окне "Loop Configuration" или терминал "Mode" для управления работой цикла. Режимы цикла "Timed Loop Scheduler" позволяют подогнать время исполнения к первоначально заданному, определить цикл как новую задачу, которая запустится в текущий момент времени, исполнить пропущенную итерацию или пропустить итерацию.

Допустим, что Вы установили период цикла "Timed Loop" 100 мс и задержку старта 30 мс. И, конечно же, надеетесь, что первая итерация начнется через 30 мс, и будет выполняться 100 мс. Суммарное время на первую итерацию составит 130 мс, а для всех последующих - по 100 мс. Но в это же время выполняются и другие циклы, идет обмен с устройствами В/В и т.п. И первая итерация Вашего цикла может быть выполнена, например, через 240 мс. А Вам хочется, и это реализуемо с помощью функций управления режимами работы цикла, "подогнать" время выполнения к уже работающему общему "расписанию" программы. И "Timed Loop" будет делать это. Естественно, насколько это возможно. Следующая итерация цикла будет выполнена к моменту времени 330 мс, и все дальнейшие будут выполняться с ранее заданным шагом 100 мс к моментам времени, определяемым рядом 430 мс, 530 мс, ... Если же подгонка времени не важна, "Timed Loop" будет выполнен немедленно, но Вы получите иной ряд: 240 мс, 340 мс, 440 мс ... Что означает запаздывание выполнения цикла? Потеря данных для следующей итерации, а еще хуже - пропуск данных, например, поступающих от АЦП.

Если "Timed Loop" пропустил две итерации или некоторые данные за этот период времени, буфер все же может содержать потерянные данные. И Вы, возможно, захотите обработать пропущенные данные во время выполнения цикла, прежде чем произойдет подгонка расписания. Но если Вы не хотите обрабатывать пропущенные данные, "Timed Loop" сможет проигнорировать их и обрабатывать только вновь поступившие. И тем не менее, при работе "Timed Loop" будут наблюдаться некоторые "шероховатости", объяснить которые можно изменением времени перестройки "расписания" всей программы.

По умолчанию "Timed Loop" игнорирует любые данные, пропущенные в течение итераций, и обрабатывает начальное "расписание". В этом режиме входной терминал "Mode" содержит символ "D", что означает игнориро-

вать, и символ "M", что означает поддержку начального "расписания".

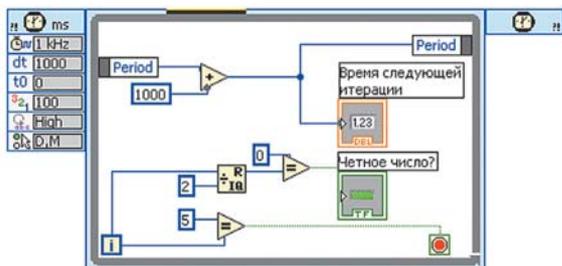
Удалите метку из пункта "Discard missed" в диалоговом окне "Loop Configuration" и Вы получите возможность обрабатывать пропущенные данные. На входном терминале появится символ "P". А для подстройки скорости работы цикла удалите метку из пункта "Maintain original phase" там же, в конфигураторе, и на входном терминале появится символ "I". Было - "D,M", стало - "P,I".

Задание режимов работы цикла "Timed Loop" можно выполнять программно через входной терминал, используя вход "Mode". Для этого следует открыть "Loop Configuration" и установить метку в пункте "Use terminal" секции "Timed Loop action on late iterations". А теперь закройте диалоговое окно и щелкните правой клавишей манипулятора на входном терминале "Mode". Из выпадающего меню выберите "Create»Constant" или "Create»Control".

### Динамическое изменение параметров и останов цикла Timed Loop

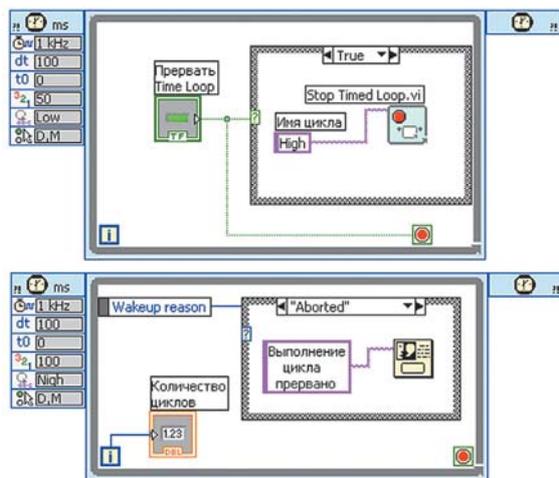
Для получения информации о работе цикла, такой, например, как запаздывание выполнения, период, смещение, следует использовать левый терминал. Подсоедините его к правому терминалу данных и у Вас появится возможность для динамического изменения этих параметров. Например, первый цикл выполняется за 1000мс, а уже следующий на 1000мс больше:

Вам, опытному LabVIEW-программисту, уже интуитивно понятно как реализовать динамическое изменение



параметров цикла "Timed Loop". Максимум три попытки и это получится. Но все же последовательность действий может быть следующей. Если в "Timed Loop" не видны левый или правый терминалы или необходимые параметры терминалов, щелкаем правой клавишей манипулятора на соответственно левой или правой границе цикла. Из открывшегося меню выбираем пункт "Show Left Data Node" или "Show Right Data Node". Затем щелкаем правой кнопкой манипулятора на терминале, выбираем "Select Item" и из выпадающего меню - нужный пункт. И последний шаг - опять щелкаем правой клавишей манипулятора по терминалу и выбираем "Create»Control" или "Create»Constant". Дальше - дело техники. Кстати, терминалы растягиваются по вертикали. Попробуйте и увидите, что получится.

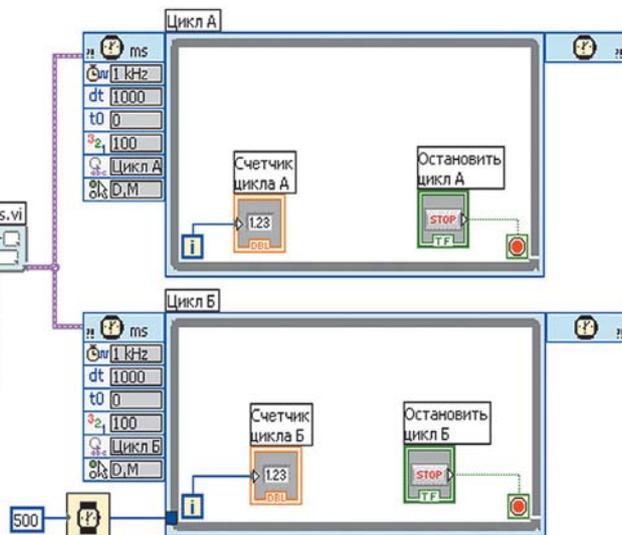
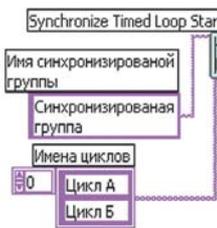
Идем далее. Как остановить работу цикла "Timed Loop"? Очень просто. Использовать "Stop Timed Loop VI" для остановки работы цикла программным путем. Укажите имя цикла "Timed Loop", который вы хотите остановить, в строковой константе и подключите ее к терминалу "Name Stop Timed loop" как показано ниже. Цикл "Timed



Loop" с именем "Low" включает в себя "Stop Timed loop". Цикл же "Timed Loop" с именем "High" постоянно выполняется, а в нем отображается число выполненных итераций. Если Вы активизируете кнопку "Прервать Time Loop" на передней панели, то на левом терминале с именем "Wakeup reason" появится сигнал "Aborted" и после Вашего подтверждения "О.К." цикл остановится. Обратите внимание: цикл "High" имеет более высокий приоритет, а цикл "Low" - больший период выполнения.

### Синхронизация циклов Timed Loop

Для синхронизации работы нескольких циклов типа "Timed Loop" предусмотрен "Synchronize Timed Loop Starts VI". Все циклы при этом имеют один источник синхронизации и одинаковые задержки. Например, в Вашей программе имеется два цикла "Timed Loop", которые должны выполняться в соответствии с таким сценарием: необходимо, чтобы первый цикл выполнялся первым и генерировал данные, а затем второй - обрабатывал данные. Для гарантированного использования одинакового времени старта циклов, следует создать константу "имя синхронизированной группы" и массив, содержащий имена циклов. А далее просто подключаем их к соответствующим входам "Synchronize Timed Loop Starts" VI. И все!



Материал подготовлен М.И.Манжелий, Инженерный Центр "Автоматизированные системы контроля", Москва, Россия.