

Halex07  
Прописка

Добавлено: Пт Окт 31, 2008 12:43 pm Заголовок сообщения:

Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

### 3.5. Вопросы по индикации и индикаторам.

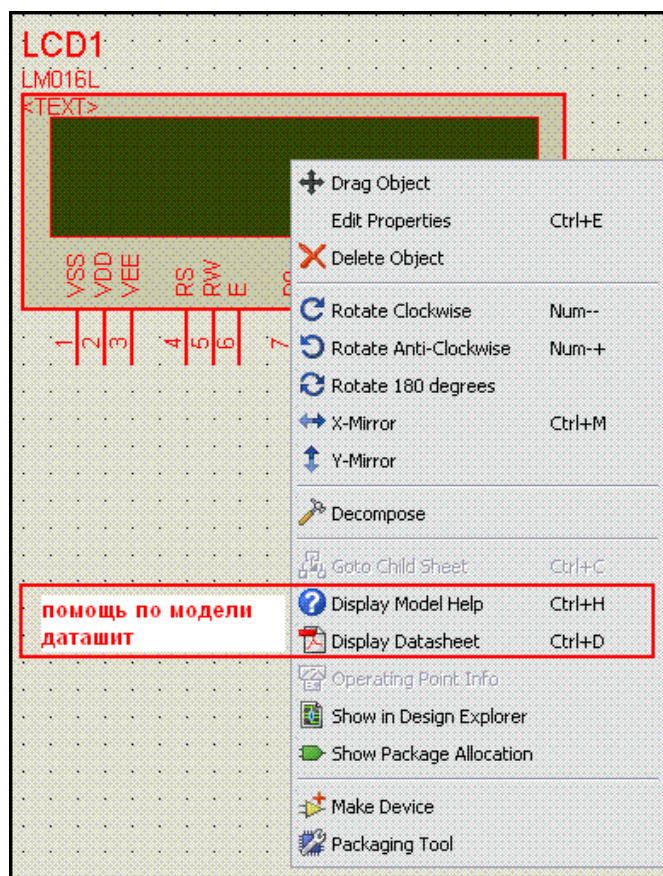
#### 3.5.1 Алфавитно-цифровые ЖК дисплеи на основе HD44780.

В Протеусе эта группа сосредоточена в библиотеке **Optoelectronics => Alphanumeric LCDs**. А соответствующая библиотека DLL носит название **LCDALPHA.DLL**. Большинство вопросов можно разделить на две группы: неправильно выводятся или не выводятся совсем символы и русский текст на дисплее.

Основная масса ошибок касающихся неправильного вывода символов возникает у тех, кто пишет программы на ассемблере, либо слепо использует чужие асм-коды и прошивки МК. Им следует помнить, что модель **HD44780**, используемого в Протеусе, написана строго по даташиту на данный контроллер, т.е. работает в соответствии с заложенными последовательностью и временами отработки команд и ответов на них. Некоторые авторы пренебрегают этим, т.к. реальные дисплеи допускают пропуски части команд в последовательности инициализации или отклонение времени их выполнения. Вот здесь чаще всего и возникают конфликты, причем «в железе» работает, а в симуляторе – нет. Рекомендация только одна – изучайте описание дисплея (Datasheet) и ищите ошибку. Тем более, что для LCD искать даташит по просторам Интернета и не потребуются. Достаточно при подключенном канале щелкнуть в ISIS правой кнопкой мыши по LCD помещенному в проекте и выбрать активный пункт **Display Datasheet**. Протеус сам предложит скачать даташит (406 кбайт английский PDF) и сохранит его для дальнейшего автономного использования в папке **Documents and Settings\All Users\Документы\Downloaded Data Sheets\**. При щелчке правой кнопкой доступен автономный файл помощи – пункт **Display Model Help**, правда довольно краткий. Описанным способом можно скачать даташиты и на другие компоненты: графические дисплеи, микроконтроллеры, цифровые микросхемы, если активен пункт **Display Datasheet**. Причем легальный или нет у вас Протеус в данном случае не важно – регистрация не требуется – главное наличие подключенного канала Интернет.

Тем, кто использует для написания микропрограмм языки высокого уровня: Си, Бейсик, Паскаль повезло несколько больше – о них, как правило, позаботились авторы используемых компиляторов. В подавляющем большинстве компиляторов присутствуют готовые библиотеки для работы с LCD. Но и здесь есть «подводные камни». По умолчанию большинство компиляторов используют матрицу 5x7, в то время как для нормального отображения некоторых русских символов, например **Щ** (щука), требуется матрица 5x10, иначе нижний хвостик буквы будет усечен и ваш дисплей станет слегка «шепелявить». Здесь уже требуется хирургическое вмешательство в соответствующие библиотеки компилятора. Необходимо там найти команду установки режима дисплея (по даташиту это Function Set) и в ней поменять бит F (2-й бит) с нуля на единицу. Для наиболее распространенного 4-х битного режима с 2-х строчным дисплеем это выглядит как команда 0x28 (матрица 5x7) а необходима 0x2C (матрица 5x10).

-- Прилагается рисунок: --



Вернуться к началу



Halex07  
Прописка

Добавлено: Пт Окт 31, 2008 11:05 pm Заголовок сообщения:



Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

### 3.5.2 Русский текст на экране LCD на основе HD44780 в Протеусе.

Теперь о русификации самой библиотеки **LCDALPHA.DLL**. Для ее перевода в русскоязычный режим в свое время одним из экс- (к сожалению) участников нашего форума и одновременно сотрудником Labcenter Electronics – Тенью была предложена утилита **charset.exe**. Утилита многократно проверена и работает. Суть ее действия заключается в следующем: вы берете указанную библиотеку из папки **MODELS**, извлекаете из нее ресурс в виде битмэп картинки, содержащей оригинальную таблицу символов и заменяете ее русифицированной таблицей, прилагаемой к утилите. Есть два замечания. Во-первых, необходимо использовать **LCDALPHA.DLL** именно от установленной у вас версии Протеуса, поскольку эта библиотека защищенная и меняется от версии к версии. Во-вторых, если вы не всегда используете русифицированные дисплеи, то не обязательно заменять оригинальную библиотеку модифицированной. Протеус имеет одну особенность, которой можно воспользоваться. Сначала он ищет доступные библиотеки и модели в папке с симулируемым проектом, а уже потом в своих библиотеках. Поэтому достаточно поместить русифицированную **LCDALPHA.DLL** в папку с вашим проектом, в котором необходим русский текст на дисплее. Ну и последнее, при написании микропрограммы используются не русские буквы, слова и фразы, а **массивы шестнадцатеричных кодов русских символов** в тех местах, где необходима такая величина и могучий... На русском сайте компании ATMEL лежит бесплатная утилита для перекодирования русского текста в коды символов вот прямая ссылка на эту программу: <http://www.atmel.ru/Binary/HD-44780.exe>

В прилагаемом ниже архиве содержится утилита **charset.exe**, русифицированная таблица **charset.bmp** и текстовый файл с инструкцией к применению.

Прикрепленный файл: [8461597.rar](#)

Последний раз редактировалось: Halex07 (Пт Окт 31, 2008 11:08 pm), всего редактировалось 1 раз

Вернуться к началу



Halex07  
Прописка

Добавлено: Сб Ноя 01, 2008 12:45 am Заголовок сообщения:





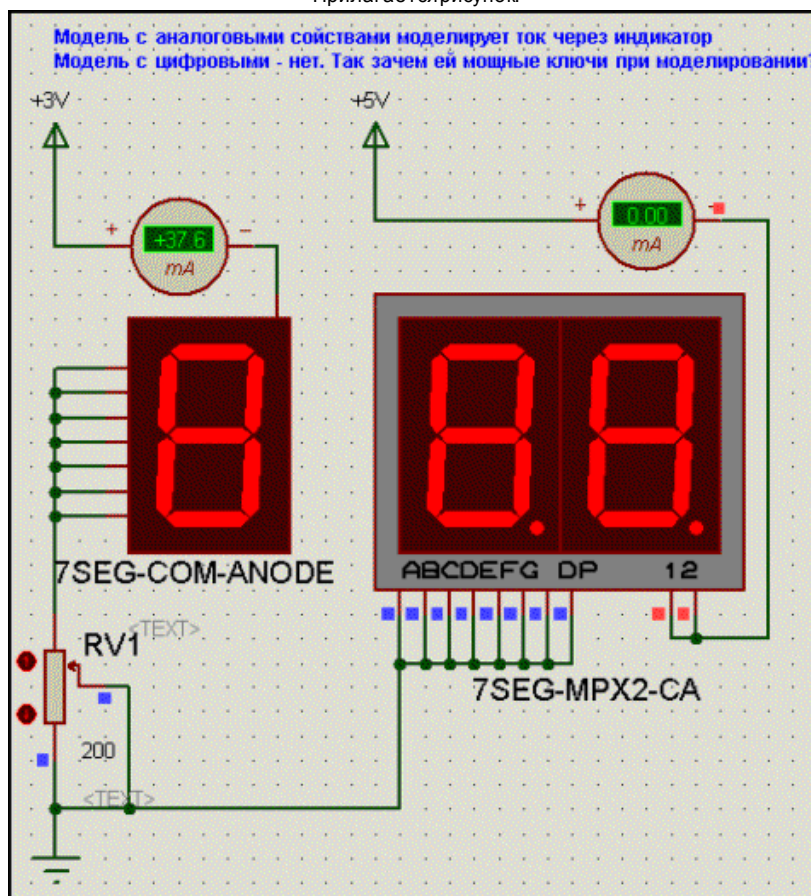
Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

### 3.5.3 Динамическая индикация на семисегментных индикаторах в Протеусе

Сразу же хочу обратить внимание всех, кто создает проекты с семисегментными индикаторами в Протеусе, что модели многоразрядных индикаторов чисто цифровые. Кто не верит – щелкните по индикатору правой кнопкой и выберите пункт **Display Model Help** (переведите первое предложение второго абзаца самостоятельно или посмотрите на приложенную картинку). Так что привешивание к ним в Протеусе всевозможных мощных транзисторных ключей, токоограничивающих резисторов и т.п. – это из разряда «аналоговых извращений» и здесь не рассматривается. Отмечу только, что это касается многоразрядных (2 и более) моделей, одnorазрядные **Schematic** модели обладают аналоговыми свойствами, поэтому если их навешать достаточно количество – возникнет перегрузка ЦПУ компьютера. И еще, поскольку многоразрядные модели цифровые – яркость их свечения не меняется, чего не скажешь о реальных индикаторах. Поэтому при проектировании реального устройства необходимо учесть следующие азы:

- 1) человеческий глаз не замечает мерцания с частотой выше 25 Гц (вспомним про эффект 25 кадров), поэтому для исключения мерцания необходима минимальная частота обновления  $25 \times N$  (количество индикаторов);
- 2) для того чтобы сохранить яркость свечения индикаторов в динамическом режиме – ведь средний ток через сегменты упадет – необходимо уменьшать токоограничивающие резисторы, а при большом количестве разрядов возможно и увеличивать напряжение питания индикаторов;
- 3) вот исходя из предыдущего пункта и вылезут на свет мощные ключевые элементы и т.п., так как засветка, например, всех восьми сегментов (учитывая точку) знакома с током сегмента от 3 до 10 мА даст суммарный ток от 24 до 80 мА, но это в статике. Средний же ток через ключ, учитывая скважность импульсов для восьмизнарядного  $N=8$  для сохранения яркости свечения грубо необходимо увеличить в 8 раз (в реальности зависимость нелинейна и несколько меньше) т.е. получим от 192 до 640 мА!!! Здесь есть над чем поломать голову. Необходимо учесть и длительность импульса и реальные импульсные характеристики конкретного индикатора по его даташиту, чтобы его не спалить. Так что Протеус в данном случае может дать результат отличный от того, что вы увидите и увидите ли в реальном устройстве.

-- Прилагается рисунок --



[Вернуться к началу](#)



Halex07  
Прописка



Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

Добавлено: Сб Ноя 01, 2008 12:51 am Заголовок сообщения:



### 3.5.4 Анализ динамической индикации с помощью цифрового (DIGITAL) графа.

В примерах (**SAMPLES**), прилагаемых к Протеус, я нашел только один, который демонстрирует характерные особенности использования динамической индикации – это **AVR Tiny15 Demo**. На нем и остановимся, т.к. в нем четко просматриваются принципы, на которые в свое время в одной из веток форума указывал **Dosikus**, за что ему отдельное спасибо. До недавнего времени эта ветка существовала по ссылке:

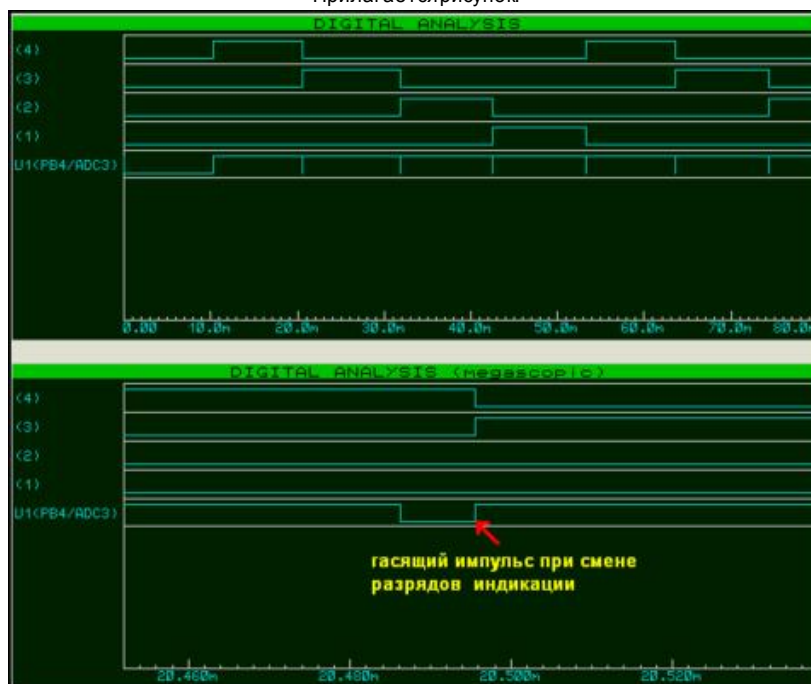
<http://kazus.ru/forum/topics/10434.html>

Кроме того, материал этой ветки с любезного разрешения автора размещен на сайте Каллиграфа:

<http://www.kaligraf.narod.ru/nedodelki.html>

Суть реализации метода заключается в том, что на момент смены разрядов индикация прерывается на короткий промежуток времени. Если этого не сделать – наблюдается наложение разряда на разряд и вместо желаемых цифр в лучшем случае будут отображаться восьмерки (все сегменты засвечены), а иногда и полная чехарда в виде отдельных горящих сегментов. Итак, копируем папку **AVR Tiny15 Demo** из **\SAMPLESVSM for AVR\** в удобное место на диске, чтобы слегка модифицировать пример. Дело в том, что в исходном примере от версии 7.2 SP6 мне не удалось смоделировать цифровой граф – Протеус ругался на модель семисегментного индикатора, а для рассмотрения особенностей динамической индикации он нам потребуется. Проблема решилась когда я удалил используемый индикатор из проекта и вновь добавил из библиотеки 7.2 SP6 (т.е. двойной щелчок по индикатору в проекте правой мышью, затем в меню **Edit** по метке **Tidy**, ну а затем добавляем из библиотеки 7SEG-MPX4CA). Навешиваем **Voltage Probe** на выводы **1, 2, 3, 4** индикатора и на стробирующий выход МК – **PB4/ADC3**. Добавляем **Graph Mode** в меню слева цифровой (**Digital**) граф в проект, добавляем его по одному **Add Traces => Probe1** установленные пробники, ограничиваем в свойствах графа время **Stop Time** на 80m или 100m (мсек) и щелкаем по пробелу (или **Simulate Graph**). Я так подробно здесь рассматриваю использование графов, чтоб не описывать их использование отдельно по ним тоже часто задаются вопросы. На приведенных на рисунке графах хорошо видно что в конце засвечивания каждого знакома (высокий уровень **4,3,2,1**) проходит гасящий строб (**PB4/ADC3**) на нижнем графе это место увеличено. Делается это следующим образом: щелкаем по графу правой кнопкой мыши выбираем **Maximize** внизу в меню на раскрывшемся графе будут дополнительные кнопки прокрутки времени (стрелки) увеличения уменьшения (4 различных лупы Zoom). Если необходимо поставить маркеры щелкаем по тому месту где он нужен (зеленый) и с нажатой **Ctrl** вторую точку (красный) внизу на черной полоске отразится их временное положение в цифровой форме и приращение (dx) между ними. Слева на оси Y отображается состояние сигнала для зеленого (основного) маркера. Итак, поигравшись с нашими графами, мы имеем следующее: суммарный период индикации 4 цифр составляет около 43 мсек (частота соответственно около 24 Гц), длительность сигнала на знакоместе около 10,5 мсек при этом в конце его следует гасящий строб длительностью около 10 мсек.

-- Прилагается рисунок --





Вернуться к началу



**Halex07**  
Прописка

Добавлено: 06 Ноя 01, 2008 1:04 am Заголовок сообщения:



Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

#### Подведем итоги нашего исследования:

- \*\*\* В классическом рабочем, прилагаемомс Протеусом, примере применен именно тот метод, который предложил в свое время **Dosikus**. И метод работаетна все 100%.
- \*\*\* Для того, чтобы избежать чехарды на индикаторах в Протеусе необходимоорганизоватьгашение индикации на период смены разрядов. Время, на которое гасится индикация может быть достаточно малым, но оно должно быть. Как его организовать– это уже прихоть разработчика. Если у вас внешние буферные микросхемы (как в рассмотренном примере) то можно стробировать их отдельным сигналом, если индикаторы подключены непосредственно к порту МК, можно, например, переводить выходы порта в высокоимпедансное состояние – фантазия тут безгранична – главное результат
- \*\*\* Использование одноразрядных аналоговых моделей индикаторов (на моем Атлоне 64 с частотой 3200 МГц свыше трех штук) приводит к 100% загрузке процессора компьютера и срыву симуляции в реальном времени.
- \*\*\* Ну и наконец главное – моделирование динамической индикации в Протеусе не дает полной гарантии работоспособности реального устройства, т.е. потребует еще и отработку на макете, но тем не менее позволит значительно сократить время разработки

Вернуться к началу



**Halex07**  
Прописка

Добавлено: 06 Ноя 01, 2008 1:06 am Заголовок сообщения:



Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

#### Список наиболее интересных на мой взгляд ресурсов и литературы, в которых рассматриваются вопросы динамической индикации:

**Книги** (ссылки ищите в соответствующей ветке форума):

- \*\*\* Вольфганг Трапперт «Измерение, управление и регулирование с помощью AVR-микроконтроллеров» (*глава 2 полностью посвящена динамической индикации, приведен расчет токоограничивающих резисторов*).
- \*\*\* В. Н. Баранов «Применение микроконтроллеров AVR: схемы, алгоритмы, программы» (*глава 4 посвящена динамической индикации*). Тот же материал опубликован в журнале «Схемотехника», № 5, 6 за 2006 г. Автор имеет свой сайт: <http://bvn123.narod.ru/>
- Он-лайн ресурсы:**
- \*\*\* А. В. Микушин Цикл лекций по теме «Цифровые устройства» на сайте СибГУТИ <http://www.sibstis.ru/~mavr/contCVT.htm> (*раздел 7.4 посвящен динамической индикации*)
- \*\*\* Динамическая индикация 9 разрядного индикатора по последовательной шине. (от DimAlt) у Радиокота <http://radiokot.ru/lab/controller/08/> (*приложен рабочий проект для Протеуса с тестовой программой для Atmega8 на WinAVR*).
- \*\*\* Динамическая индикация и регулировка яркости <http://arv.radioliga.com/index.php?ent&task=view&id=101&Itemid=49> (*интересная идея по ШИМ регулировке яркости индикаторов от Романа Абраша автора цикла статей по МК в журнале «Радиолюбитель»*)

Вернуться к началу



**Halex07**  
Прописка

Добавлено: 06 Ноя 02, 2008 1:31 am Заголовок сообщения:



Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

#### 3.6. Подробнее о графах (или графиках, если кому-то нравится такое название). 3.6.1. Что можно сделать с помощью графов?

Мой ответ однозначный - практически все, даже тогда, когда реал-тайм симуляция бессильна или у вас старенький "ПЕНЬ", а не суперкомпьютер. В частности HELPe авторами программы упоминаются такие исследования как анализ слабых сигналов, шумов и параметрический анализ. Я не знаю, почему так незаслуженно обходится стороной их применение у русскоязычных пользователей Протеуса. Те, кто работает этой программой не первый год, наверно давно раскусили возможности этой опции ISIS, но помалкивают о подробностях. Попробую приобщить начинающих к применению графов в своих разработках. Графы по своей сути очень близки к запоминающему осциллографу, плюс к тому в отличие от интерактивного осциллографа, который в седьмой версии не лишен некоторых "глюков", дают более реальную картинку происходящего процесса. К тому же в отличие от четырехканального осциллографа на графе можно расположить большее количество одновременно наблюдаемых сигналов. Мы уже сделали это пунктом раньше, расположив 5 наблюдаемых каналов в цифровом графе и это не предел! Итак, вопросу применения графов для анализа посвящен раздел **GRAPH BASED SIMULATION** в **ProSPICE HELP** (он же **PROTEUS VSM HELP**) - многое там понятно по картинкам и без знания английского. Кроме того, в **Samples** к Протеусу есть отдельная папка **Graph Based Simulation**,

полностью посвященная примерам с использованием графов. Заниматься подробным переводом HELP-а я не имею ни возможности, ни желания, но остановлюсь на некоторых ключевых моментах.

### 3.6.2. Какие графики можно симулировать в Протеусе? Примеры использования.

Эти вопросы рассмотрены в разделе **ANALYSIS TYPES** того же **ProSPICE HELP**. Остановлюсь на самых употребимых, остальные попытаюсь дать только ссылки на примеры в Протеусе.

Щелкнув по соответствующему значку в левой панели, мы получим меню для выбора типа графика, размещаемого в проекте (всего их 13 типов). Наиболее употребимые в обиходе (с моей точки зрения) это: **ANALOGUE** (аналоговый), **DIGITAL** (цифровой), **MIXED** (смешанный). А в последнее время я все чаще стал их эксплуатировать **INTERACTIVE** - интерактивный график и **CONFORMANCE** - график соответствия. На них чуть ниже остановимся особо. Остальные:

**FREQUENCY** - частотный анализ, **NOISE** - анализ шумов, **DISTORTION** - анализ искажений. Смотрите в примере 741.DSN. При запуске анализа будут вылетать "горчичники" об отсутствии Pin1 и Pin2 на дочернем листе - это соответственно выводы коррекции 1 и 5, которые в дочернем листе не прорисованы. Для всех этих графиков в свойствах (Properties) аналогично: Reference - источник, Start и Stop Frequency - начальная и конечная частоты исследования, Interval - шаг по оси X, No Steps/Interval - количество шагов исследования на интервал.

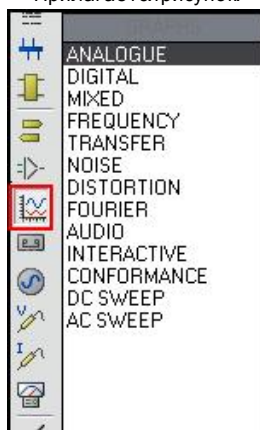
**TRANSFER** - анализ передаточных характеристик. Смотрим одноименный TRANSFER.DSN. Ба, да это готовый справочник по импортным транзисторам. Да еще в стиле старых добрых советских справочников, только меняй тип транзистора из библиотеки. В современных этих графиков не найдешь. Я думаю, в назначениях свойств разберетесь самостоятельно, или с помощью старого справочника по транзисторам, где помещались подобные передаточные характеристики.

**DC SWEEP** и **AC SWEEP** - создают соответственно график изменения напряжения (тока) и семейство частотных характеристик в зависимости от изменения какого либо параметра компонента. Смотрите в примере SWEEP.DSN. Для DC показана зависимость изменения тока через диод от приложенного напряжения, а для AC семейство частотных характеристик RC-цепочки при изменении R. В свойствах графа также довольно легко понять что и как меняется.

**AUDIO** и **FOURIER** - как и следует из названия соответственно анализируют звуковые сигналы и спектральный анализ сигнала по Фурье. Смотрите пример их применения в FOURIER.DSN.

Великолепный классический пример синтеза прямоугольного сигнала из ряда синусоидальных сигналов.

-- Прилагается рисунок --



Вернуться к началу



Halex07  
Прописка

Добавлено: Вс Ноя 02, 2008 1:40 am Заголовок сообщения:

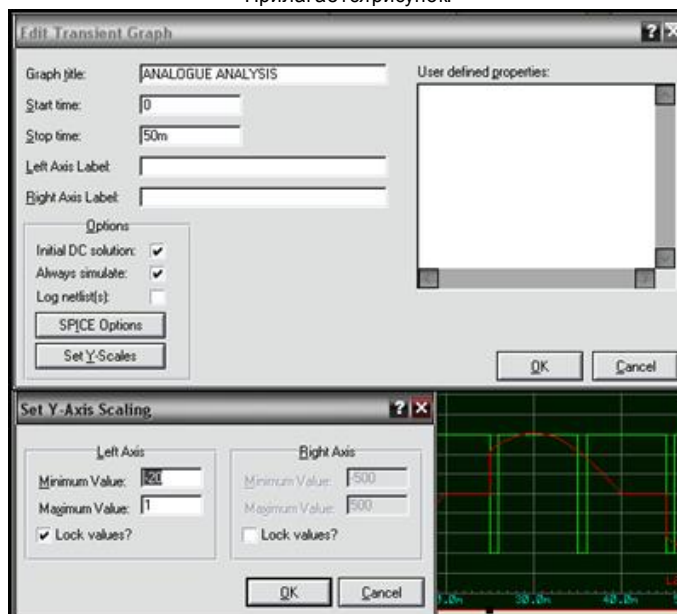


Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

Теперь подробно о моих излюбленных:

**ANALOGUE** - сродни обычному запоминающему осциллографу. На нем можно анализировать как аналоговые, так и цифровые сигналы. Графики различных сигналов будут различаться цветом, но при этом накладываться один на другой. В свойствах графа (см. картинку) можно задать начальное (**Start Time**) и конечное (**Stop Time**) время анализа. В **Left** и **Right Axis Label** можно ввести названия, которые будут показываться у соответствующих вертикальных шкал. Если щелкнуть по кнопке **Set Y-Scales**, можно установить фиксированные значения для верхнего и нижнего предела каждой шкалы, установив галочку **Lock Value**. Для цифрового графа эта функция будет неактивна.

-- Прилагается рисунок --



Вернуться к началу



Halex07  
Прописка

Добавлено: Вс Ноя 02, 2008 1:48 am Заголовок сообщения:



Зарегистрирован:  
May 03, 2007  
Сообщения: 191  
Откуда: Владимир

**DIGITAL** - практически заменяет многоканальный логический анализатор. В отличие от последнего позволяет наблюдать сигнал непосредственно на многопроводных шинах. Для этого на шину устанавливается V-пробник с именем, соответствующим лэйблу шины. Естественно на шину предварительно как на провод вешается **Label**, но наименование у него должно быть несколько отличное от отдельных проводов. Например в шину входят 4 провода с именами от **D0** до **D3**. Тогда установленный **Label** должен иметь название **D[0..3]** и такое же значение присваивается пробнику. Обратите внимание диапазон охвата в квадратных скобках должен соответствовать количеству проводов в шине и между цифрами ставится не тире, а точка, как принято у нас, а **ДВЕ** точки. При этом на графе в стабильные по состоянию промежутки времени высвечивается цифровое шестнадцатиричное значение шины (сиреневая трасса).

-- Прилагается рисунок --

