



ISaGRAF- важные мелочи!

*Рекомендации подготовлены сотрудниками
отдела программного обеспечения компании
"ХОЛИТ Дэйта Системс, г.Киев"*

Во время работы все мы сталкиваемся с массой мелких проблем, не настолько серьезных, чтобы тратить на них неделю, но отнимающих пару-тройку часов. Мы собрали в этой статье несколько заметок, описывающих решение будничных задач. Возможно, кому-то они покажутся бесполезными, а для кого-то сэкономят два часа времени. Итак....

Лицензионная политика ICS Triplex ISaGRAF

Применение лицензионного программного обеспечения в домашних условиях - вопрос личной философии. Однако в коммерческих проектах личность уступает государству, которое явно требует использовать исключительно легальное ПО. Такой подход вполне правомерен - Вы зарабатываете деньги с использованием тех или иных инструментов, так почему их разработчик должен оставаться в стороне?

Хочется отметить, что процент тех, кто предпочитает не платить, все уменьшается. Прежде всего это вопрос удобства - купив лицензионный продукт, Вы получаете техническую поддержку. Однако оставив размышления о преимуществах использования легального ПО юристам, перейдем к вопросам более практическим - лицензионной политике компании ICS Triplex ISaGRAF.

Программные продукты ICS Triplex ISaGRAF имеют два вида защиты от копирования - аппаратный и программный. Эта защита не является чем-то особенно новым, однако содержит несколько особенностей настройки, которые мы и опишем.

Лицензированию подлежат количество точек ввода/вывода и дополнительные возможности, характеризующие работу самого приложения - будет ли приложение создаваться для мультизадачной среды, когда на одном физическом контроллере параллельно работает несколько программ, и следует ли выполнять код на распределенных вычислительных ресурсах (схема: одно приложение - много контроллеров)

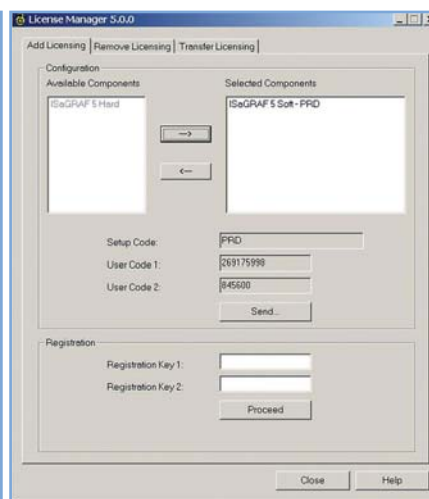
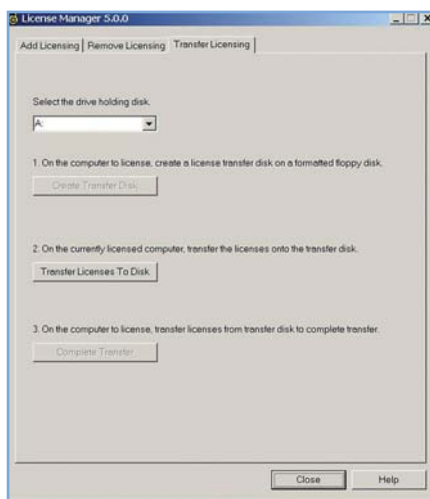
Аппаратная защита строится на применении специального ключа, так называемого dongle, который устанавливается в USB или LPT порт компьютера. Ключ содержит в себе информацию, какие именно возможности можно сделать доступными для пользователя. Эта информация считывается во время компиляции проекта и, если Вы превысили разрешенный уровень, программу собрать не удастся.

Помимо самого ключа нужно установить драйвера и программное обеспечение, позволяющее работать с аппаратной защитой из приложения (в общем случае из Workbench). В настоящий момент компания ICS Triplex ISaGRAF применяет ключи Sentinel принадлежащие компании SafeNet Inc. Процедура активации ключа достаточно тривиальна - нужно загрузить архив, содержащий инсталляционные файлы, произвести установку и перезагрузить компьютер. Все, Вы готовы работать с полнофункциональной

версией ISaGRAF Workbench. Ключ можно извлекать после окончания работы, но перед ее началом (перед включением компьютера) он должен быть установлен обратно, чтобы менеджер лицензий обнаружил его и считал всю необходимую информацию.

Наиболее свежее программное обеспечение, необходимое для работы с ключами, можно загрузить как с сайта производителя (<http://www.safenet-inc.com/support/tech/sentinel.asp>), так и с отечественного (<http://www.isagraf.com.ua>).

Программная защита построена по методу активации. Это значит, что лицензия выдается на отдельный компьютер и привязывается к конкретному аппаратному обеспечению. Удобное решение для производителя ПО, но достаточно хлопотное для конечных пользователей, ведь достаточно часто бывает, что компоненты рабочего компьютера меняются, да и Windows не является образцом надежности.



После установки программного обеспечения необходимо получить регистрационные ключи. Запустите Менеджер Лицензий (Licensing) из группы ICS Triplex в установочном меню и в первой вкладке выберите необходимые Вам компоненты.

В случае, если Вы являетесь покупателем ICS Triplex ISaGRAF, коды из графы User Code 1, User Code 2, а также Setup Code следует отправить нам для получения регистрационных ключей.

В нашей бурной жизни разработчику часто приходится менять рабочее место - и хорошо, если это происходит в результате улучшения характеристик ПК, а не по причине поломки. Если Вы решили установить ISaGRAF на другом компьютере, старые лицензионные коды уже не подойдут и потребуются создавать новые. Перенос лицензии разбит на несколько шагов и потребует одной чистой дискеты и небольших затрат времени. После того как на новый компьютер были установлены все необходимые инструменты ICS Triplex ISaGRAF, необходимо создать трансферный диск. Сделать это можно, запустив Менеджер лицензий, перейдя в закладку Transferring и нажав кнопку Create Transfer Disk.

Программа соберет сведения о вашем ПК и сохранит их на дискете. После того как эта процедура будет завершена, необходимо перенести уже существующую лицензию со старого компьютера на дискету. Делается это все в том же меню Transferring, разве что на другом ПК. Установив носитель в привод, нажмите кнопку Transfer Licensis to Disk. Эти действия приведут к тому, что оригинальная лицензия будет перенесена на диск и удалена на данном компьютере. В результате этих действий Вы становитесь обладателем достаточно ценной дискеты и двух нелегальных продуктов ISaGRAF. Установив дискету в новый компьютер, перейдите в закладку Transferring и нажмите на кнопку Complete Transfer. Все, процедура завершена, лицензируемые компоненты станут активными и Вы можете приступать к работе на новом месте!

Отладка приложений в ISaGRAF 3

ISaGRAF представляет собой систему разработки управляющего программного обеспечения для промышленных объектов. Это значит, что у разработчика нет права на ошибку в таких программах и перед запуском

они должны быть самым тщательным образом проверены и отлажены. Следует помнить, что в реальности система управления будет состоять как минимум из двух компонентов - логики приложения, построенного на ISaGRAF, и средства отображения и обработки информации (это может быть SCADA-система или разработанное самостоятельно ПО, сейчас это не имеет значения). Все компоненты должны быть подвергнуты тестированию, как по отдельности, так и совместно. Мы не будем приводить стратегий тестирования, так как каждый разработчик придерживается своих привычных правил, а расскажем, какие инструменты предоставляет ISaGRAF в помощь инженеру.

Третья версия ISaGRAF на настоящий момент получила в нашей стране наибольшее распространение, главным образом, благодаря ее использованию в контроллерах ICP_DAS. Эта простая и доступная среда разработки имеет два инструмента, которые помогут Вам в отладке. Это непосредственно режим Симуляции и утилита Прожектор.

Вы только что создали приложение. Оно активно работает с платами ввода/вывода и проводит на основе измерений некоторые вычисления. Нужно проверить, правильно ли реализован алгоритм, как он себя поведет с различными комбинациями входных параметров. Скомпилируйте проект и запустите его в Отладчике.

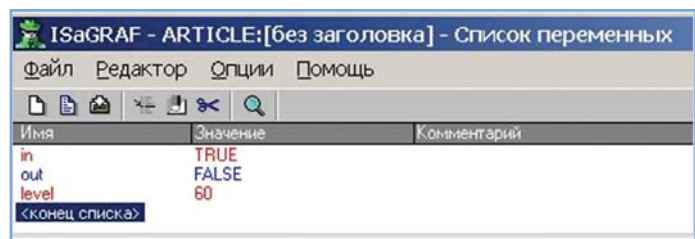
Появившееся окно будет содержать все присоединенные платы ввода/вывода, причем значения сигналов на платах, отвечающих за входные сигналы, можно менять. Изменения, внесенные через этот интерфейс, автоматически попадают в программу, меняя ее поведение в зависимости от заложенного алгоритма. Проведите эксперимент, задавая граничные значения или же уровни, которые никогда, казалось бы, не должны возникнуть в реальной жизни. К сожалению, лабораторные условия сильно отличаются от производственных и нужно быть готовым ко всему. Еще одна дополнительная проверка лишней не будет.

Просмотреть, как именно распространяются сигналы в самой программе, какие значения принимают выходы

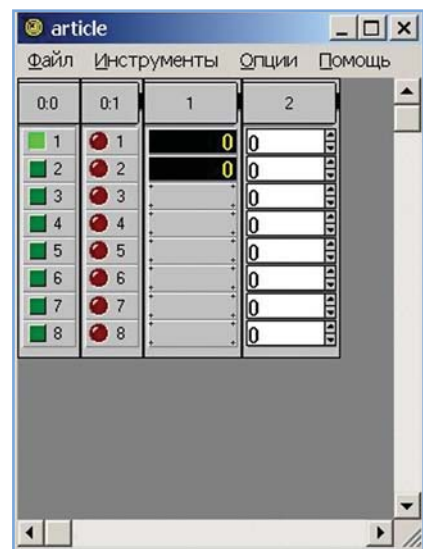


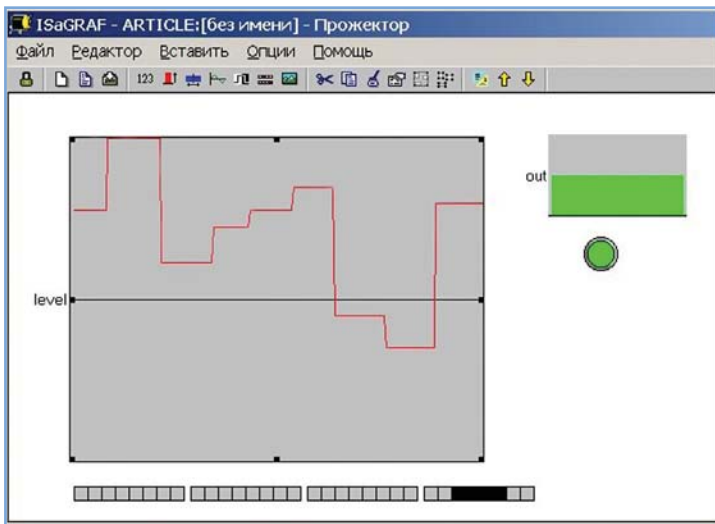
модулей после обработки Вы можете, открыв исходный текст программы во время симуляции. Редактор позволит Вам модифицировать значение любой переменной, для этого достаточно сделать двойной щелчок и задать новое значение прямо во время выполнения программы. Оцените результат.

Не всегда удобно смотреть изменение данных в окне редактирования текста, да и не все используют в качестве основных графические языки программирования. Описанный выше метод не подойдет для отладки программ, созданных с использованием ST или IL. Однако и тут ICS Triplex ISaGRAF предлагает наглядное решение. Воспользуйтесь Списком Шпиона. За этим громким названием скрывается маленькая, но очень полезная утилита, позволяющая отображать изменения в переменных во время выполнения программы. Утилиту можно вызвать после того, как



проект запущен на симуляцию. В окне Отладчика ее можно найти в меню Инструменты->Списки шпиона. Работать со списками очень просто: нажмите на кнопку для добавления новой переменной и в появившемся диалоге выберите те переменные, с которыми есть необходимость работать. Переменные можно не только просматривать, но и менять их значения во время выполнения программы. Для того,





чтобы в дальнейшем вернуться к отладке и не создавать список заново, сохраните его под каким-нибудь именем, а затем, при необходимости, откройте его через меню Списка шпиона. К сожалению, Список имеет ограничение на количество задаваемых переменных и Вы можете одновременно отслеживать состояние только 32-х переменных.

ISaGRAF не позволяет строить красивые пользовательские интерфейсы, его область применения несколько иная. Однако получить простейший интерфейс, который позволил бы более наглядно отобразить информацию во время отладки или выполнения программы, можно и в ISaGRAF. Утилита Прожектор позволит в графическом режиме отобразить ход выполнения программы и состояние переменных. Кроме того, Вы можете менять значение переменных "на лету", проводя тестирование и внося элемент неожиданности (или реальной жизни) в размеренный ход программы.

После того как проект запущен на симуляции или выполняется на контроллере, перейдите в меню Инструменты окна Отладчика и выберите Прожектор. Открывшееся окно представляет собой рабочую область, на которой должны быть размещены элементы управления и индикации. Используя Прожектор, Вы можете создавать текстовые поля, однополярные и биполярные колонки, строить несложные графики для переменных, работать с картинками и булевыми иконками.

Работать с Прожектором достаточно просто - выделите элемент управления, который Вы хотите использовать, и разместите его на рабочей области. Двойной щелчок мышки

на нем вызовет окно привязки переменной проекта к элементу управления. Текстовые поля можно использовать для отображения числовой информации - результатов измерений или преобразований. Если Вас интересует как изменяется величина во времени, привяжите ее к графику. Этот элемент отображает последние 200 состояний переменной. Графики можно строить не только по численным значениям, но и по логическим.

Колонки являются аналогом диаграммы, но с одним столбцом. Ее заполненная область сигнализирует об уровне величины. Биполярная колонка может принимать отрицательные значения.

Битовое поле позволит показать, какой из битов числа установлен в 0 или 1.

Булевская иконка представляет собой элемент управления с двумя состояниями. Она меняет свой внешний вид в зависимости от состояния переменной, подключенной к ней. При конфигурации следует указать файлы с изображениями, которые будут символизировать значения TRUE и FALSE. Проект Прожектора можно сохранить, задав имя, чтобы потом вернуться к нему.

Как видите, у ISaGRAFa достаточно инструментов, чтобы произвести качественную отладку приложения. Вопрос "что же использовать?" зависит полностью от разработчика и его предпочтений.

Обновление прошивки ISaGRAF на контроллерах ICP DAS серии i-7188

Обновлять или не обновлять? Если да, то зачем и как?

Программное обеспечение пишется людьми, которые тоже совершают ошибки. Ошибки есть везде, они могут быть не всегда заметны, их наличие может не носить трагических последствий, или же, как самый край-

ний случай, их могут выдавать за особенности функционирования программы. Но все это не отменяет факта их присутствия. Если производитель своих продуктов ответственно, то через определенное время выходит "заплатка" или же более новая версия программы с учетом исправлений.

Конечно, кроме ошибок существует еще одна причина, по которой можно, а иногда даже и нужно, проводить обновления. Если стоит задача подключить новое устройство ввода/вывода к более старому контроллеру, Вам, скорее всего, нужно будет обновить прошивку на нем. Новые версии прошивок содержат драйвера для исполнительного ядра ISaGRAF, что позволит Вам создавать программы в ISaGRAF Workbench, взаимодействующие с этими устройствами.

Итак, причины ясны - осталось выяснить, как же обновить прошивку на контроллере. Первое, что Вам потребуется, это загрузить последнюю версию обновлений. Ее можно найти на сайтах www.icpdas.com и www.isagraf.com.ua.

Кроме прошивки Вам понадобится утилита **7188xw.exe**, которую можно найти на прилагаемом к контроллеру компакт-диске. Если все же не удалось найти такую утилиту - свяжитесь с официальным партнером фирмы.

После того, как все необходимые файлы получены, сохраните их все в одной директории, например, `c:\updates`. В ней должны быть следующие файлы: **7188xw.exe**, **7188xw.ini**, **xb030516.img** (это образ операционной системы, название может отличаться), **isa7188.exe** (исполнительное ядро ISaGRAF, название может отличаться), **autoexec.bat**.



Процесс обновления идет в два этапа и, как это не покажется странным, займет один день. Такие сроки вызваны тем, что компания ICP_DAS вынуждена выдавать индивидуальные лицензии на использование целевой функции ISaGRAF под каждый контроллер. Все устройства имеют уникальный номер, на основании которого и будет создана лицензия.

Подключите контроллер к компьютеру и запустите утилиту **7188xw.exe**. Возникший черный экран будет сигнализировать о том, что программа успешно запустилась и пытается установить соединение с контроллером. К этому времени компьютер и контроллер уже должны быть соединены через последовательные порты. Замкните контакты INIT* и GND на контроллере - подготовьтесь к этому заранее, чтобы заблокировать загрузку autoexec.bat, и включите питание. Если все было сделано правильно, в рабочей области программы 7188xw.exe появится командная строка. Чтобы узнать серийный номер контроллера, наберите команду **ver**. В результате на экране будет напечатано приблизительно следующее:

**ICP_DAS MiniOS7 for I-7188x Ver. 1.01
build 006, Feb 05 2002 10:52:09
SRAM:512K, FLASH MEMORY:512K
Serial number= 09 79 70 63 03 00 00 53**

Эта информация нужна для того, чтобы получить лицензию на исполнительное ядро ISaGRAF. Вы можете связаться с представителями ICP DAS напрямую на service@icpdas.com и запросить у них необходимую помощь, или же предоставить это нам - эксклюзивному дистрибутору ICP_DAS в Украине. В любом случае результатом будет являться файл **licence.pia**. Но перед тем как загрузить его в контроллер, необходимо обновить операционную систему. Для этого установите связь с контроллером с помощью утилиты 7188xw.exe и введите в командной строке:

i7188xw> upload

Нажмите комбинацию клавиш ALT+E и введите имя файла, хранящего образ операционной системы (в рассматриваемом случае это будет **xb030516.img**).

Запишите изменения:

7188xw> bios1

Эта операция займет максимум одну минуту. После того как она будет завершена, удалите старую версию операционной системы и ядра ISaGRAF:

i7188xw> del

Обычно загруженных файлов два и программа запросит подтверждение на их удаление. Согласитесь с ней, нажав Y.

Для того чтобы после включения питания проект ISaGRAF запускался автоматически, перепишите на контроллер файл autoexec.bat из рабочей директории c:\uploads. Для этого наберите

i7188xw> upload

Далее нажмите ALT+E и введите имя файла: autoexec.bat. Теперь следует загрузить само исполнительное ядро ISaGRAF:

i7188xw> upload

Далее уже известная Вам комбинация ALT+E и имя файла: isa7188.exe.

По прошествии минуты файл будет загружен в контроллер. Проверьте, правильно ли прошла загрузка, и соответствуют ли размеры файлов оригиналам, введя команду:

i7188xw> dir

Установите скорость работы COM-порта, по которому ISaGRAF будет принимать входящие соединения:

i7188xw> isa7188 *b=N

Параметр N может принимать следующие значения: 1=2400, 2=4800, 3=9600, 4=19200 (по умолчанию), 5=38400, 6=57600, 7=115200, 8=300, 9=600, 0=1200.

Все - можете закрыть программу 7188xw.exe, отключите питание от контроллера и разомкните INIT* GND. Теперь настало время установить полученную лицензию (файл licence.pia) на контроллер. Файл лицензии представляет собой архив проекта ISaGRAF, который не нужно компилировать или редактировать, достаточно лишь загрузить его в контроллер. Чтобы сделать это, включите сам контроллер, а на компьютере запустите ISaGRAF Workbench. Перейдите в меню **Инструменты->Архив** и выберите пункт **Проекты**. Укажите местоположение файла licence.pia и когда он появится в правой панели, выделите его и нажмите кнопку **Восстановить**. Закройте диалоговое окно и перейдите к списку проектов. Откройте проект **licence**, проверьте параметры соединения с контроллером и загрузите в него проект. Возможно, перед началом загрузки потребуются ввести пароль. Если не поступало дополнительных данных, то по умолчанию в качестве пароля выступает серийный номер контроллера.

Как только загрузка будет завершена, приложения, созданные в

ISaGRAF, будут нормально запускаться на контроллере, и Вы сможете приступить к нормальной работе.

Разработка распределенных приложений

Вы никогда не задумывались над общей архитектурой приложений, построенных с использованием ISaGRAF или же других SoftPLC решений? Во главе всегда стоит контроллер, который взаимодействует с различными УСО, локальными или удаленными. Архитектура "один контроллер - одно приложение" существует уже давно и успешно справляется со своим кругом задач, позволяя быстро и просто решать очень многие проблемы автоматизации. Но далеко не все задачи линейны, и в некоторых приложениях возникает необходимость в обмене данными с другими контроллерами. Подобная задача относится к классу распределенных вычислений, где было написано огромное количество кода и защищена не одна сотня диссертаций. Большое количество готовых продуктов и подходов иногда вызывают оторопь и недоумение - что же выбрать? Тут уже все зависит от опыта и предпочтений самого разработчика и, достаточно часто, от советов.

Попробуем предложить решение подобной задачи с инженерной точки зрения в рамках технологии ISaGRAF. Что нужно, чтобы создать распределенное приложение в ISaGRAF? Вам потребуется целевая функция под используемую операционную систему (демоверсия находится по адресу info@isagraf.com.ua). Так как приложение будет все-таки распределенное, пусть и игрушечное, потребуются два компьютера, связанных через Ethernet или последовательные порты. Конечно, можно обойтись и одним компьютером, но ведь это будет не так интересно!

Скопируйте файлы целевой системы ISaGRAF на каждый из компьютеров. После этого можно запустить ядро на выполнение командой ISaGRAF.exe. Появившийся набор консольных приложений будет свидетельствовать о том, что целевая система успешно загрузилась и готова к работе. Ядро включает в себя несколько самостоятельных приложений, которые выполняются одновременно. Это сетевой сервер ETSP, ядро виртуальной машины и менеджер. ISaGRAF имеет такую внутреннюю организацию, что подключить новый

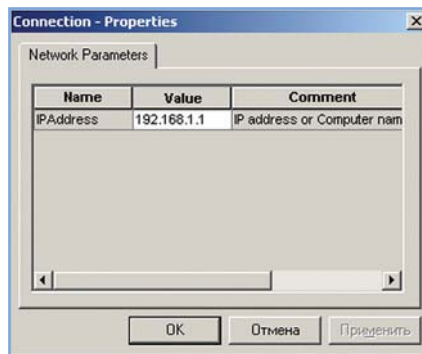
протокол обмена данными не составляет труда и в рамках одного проекта может происходить обмен по Ethernet, RS-485, CAN и по другим сетям. Пример, который хотелось бы показать, можно будет элементарно изменить для работы через последовательный порт, причем код самого приложения останется неизменным. Но все это, как говорится, лирика, пора переходить к практике!

Итак, запускаем ISaGRAF. Создаем новый проект. По умолчанию это будет Моноконфигурация (один контроллер, один ресурс). Давайте ее немного расширим. Целевые системы ISaGRAF могут работать на разных операционных системах, а иногда и вообще без них и поддерживать связь с внешним миром по всевозможным каналам связи. Например, целевая система под DOS может иметь только последовательный интерфейс, а вот под Linux Вам становятся доступными и RS, и Ethernet. Среде разработки нужно знать эти подробности, чтобы успешно подключиться к контроллеру. Вся информация о подключениях (и не только о них) содержится в файлах **TDB** (Target Definition File), которые нужно подключить к проекту перед началом работы. Для этого следует перейти в меню **File->Import->Pc definition** и выбрать файл описания (он будет находиться в директории Cmds целевой системы).

Теперь давайте укажем, что первая Конфигурация будет выполняться именно на той целевой системе, которая только что была успешно установлена. Перейдем в вид **Hardware Architecture**, и изменим свойства нашего первого контроллера, щелкнув по нему правой клавишей мыши. В меню выберем пункт Properties и изменим целевую систему на ту, описание которой только что было установлено. Это будет WIN32-TGT.

Окно **Hardware Architecture** служит также для настройки коммуникаций между контроллерами или компьютерами верхнего уровня. Вы видите, что в проекте сейчас доступна шина ETCP, горизонтальной линией разделяющая экран. В зависимости от сетевой маски, таких шин может быть и несколько. Например, часть контроллеров может работать в подсети 192.168.1.0, а другая в 10.10.0.0. Также Вы можете добавить новую физическую сеть передачи данных. Меню Insert->Network приведет Вас к диалогу конфигурации новой сети. В поставке по умолчанию их две - Ethernet и ISARSI (последовательный интерфейс).

Однако вернемся к нашему контроллеру и настроим его для работы в сети Ethernet. Вы видите, что от окна с конфигурацией отходит канал. Сделайте по нему двойной щелчок и введите IP адрес первого контроллера -



читай, компьютера, где запущена первая целевая система. Теперь добавим еще один контроллер (Конфигурацию) в проект. Меню Insert, пункт Configuration и новый пустой контроллер готов к работе! Можете сразу задать целевую систему для него, хотя можно сделать это и позже. По умолчанию контроллер создается без единого Ресурса, то есть без каких либо программ или переменных. Исправим

это, щелкнув по второй Конфигурации правой кнопкой мыши и перейдя в **Insert Resource**.

Настроим связь между контроллером и внешним миром. Щелкните по шине ETCP и удерживая клавишу, подведите канал ко второй Конфигурации. В появившемся диалоге можно сразу задать IP адрес. Отлично, сами контроллеры готовы к работе, переходим к самой интересной и одновременно самой простой задаче - собственно, созданию распределенного приложения.

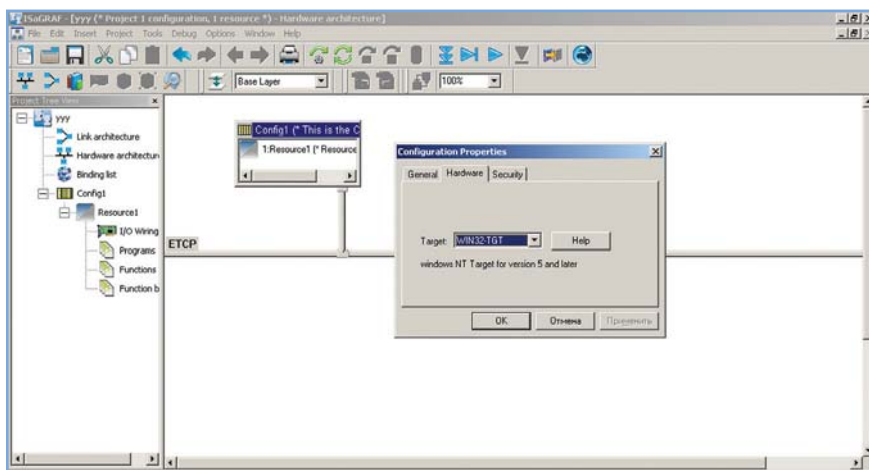
В идеологии ISaGRAF распределенное приложение должно работать с удаленными, расположенными на другом контроллере, переменными или объектами также свободно, как-будто они присутствуют локально. В ее рамках предполагается, что один из участников обмена будет выступать Поставщиком данных, а второй (и/или третий, четвертый, пятый...) - Потребителем. Безусловно, одна конфигурация может быть одновременно как Потребителем, так и Поставщиком данных.

Создадим совершенно тривиальное приложение, в котором один контроллер будет изменять переменную, а второй будет видеть эти изменения в реальном масштабе времени. Раз речь пошла о переменных, давайте создадим их. Перейдите в

Dictionary. В нем Вы увидите созданные ранее конфигурации. Развернем Ресурс 1 и перейдем в All Varibales. Перед Вами появится словарь со списком всех доступных переменных. Щелкнем по пустому месту в таблице и в появившемся диалоге или полях ввода укажем название и тип переменной. Остановимся на переменной типа DINT и назовем ее plc1. Теперь создадим переменную во второй конфигурации. Название может не совпадать, а тип следует задать такой же, то есть DINT.

Перейдем в режим **Link Architecture**. Чтобы было интереснее, давайте заставим первый контроллер выполнять какие-нибудь действия над переменной plc1, например, инкрементировать ее на каждом шаге цикла. Для этого создадим в Ресурсе 1 простенькую программу на ST. Правый щелчок мыши по пункту Programs приведет Вас в меню, где можно выбрать язык разработки для будущей функции. Дабы не усложнять изложение, воспользуемся ST. В редакторе введем код программы:

```
plc1:=plc1+1;
```





Процесорні плати

Prometheus

PC/104 CPU 486-DX2, 100MHz



- 32MB RAM на платі
- 10/100Mbps Ethernet
- 4xRS232, 2xUSB
- 1xFDD порт, IDE, IrDA порти
- 16-ліній 16-біт АЦП
- 4-ліній 12-біт ЦАП
- 4-ліній 12-біт ЦАП
- 24 цифрові лінії В/В
- 2 таймера/лічильника



Morpheus

PC/104 CPU Celeron 400-650 MHz



- До 512MB DIMM
- 10/100Mbps Ethernet
- S3 Savage 4 Chipset з 3D/2D
- IDE порт UDMA-33
- IDE порт UDMA-33
- 1xRS-232
- 1xRS-232/422/485
- 2xUSB 1.1
- LPT та FDD порти
- CompactFlash



Athena

PC/104 CPU VIA Eden 400-660MHz



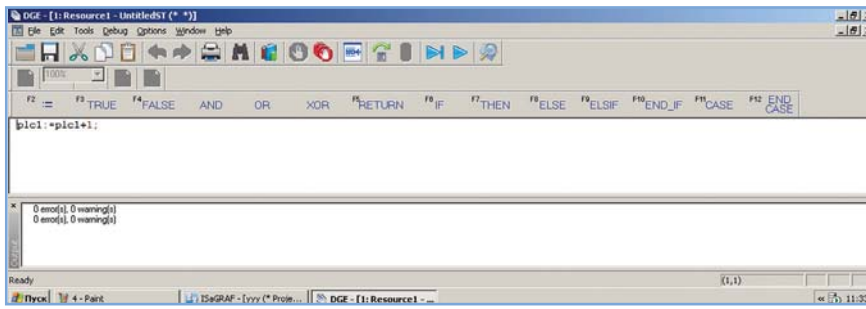
- 128MB на платі
- 10/100Mbps Ethernet
- S3 Savage 4 Chipset з 3D/2D
- IDE порт UDMA-33
- 4xRS-232, 4xUSB 1.1
- PC/104 ISA шина
- 16 аналогових ліній 16-біт АЦП, 100кГц
- 4 аналогові лінії 12-біт ЦАП
- 24 цифрові лінії В/В



Poseidon

1.0GHZ - 2.0GHZ EPIC SBC

- 256MB або 512MB 533МГц DDR2 RAM
- 10/100/1000 Ethernet
- 4xUSB 2.0, 4xRS232, 2 з RS422/485
- S-ATA та UDMA-100 IDE
- CRT та 24-біт LVDS
- 32 ліній В/В 16-біт, АЦП, 250кГц
- 4 аналог. ліній В/В 12-біт, ЦАП
- 24 цифрові лінії В/В



Логика создана, теперь давайте установим связи между контроллерами. В режиме Link Architecture выберите Ресурс 1 и нажмите на кнопку Data link(правый верхний угол), после чего перетащите связь на Ресурс 2. В результате между ресурсами должна возникнуть линия, указывающая направление связи. Двойной щелчок по этой линии приведет к появлению диалога настройки обмена переменными. Сделайте щелчок по полю From. Диалог, который сейчас появился, позволяет настроить все, что связано с обменом данными с удаленными объектами. Давайте укажем, что переменной-поставщиком будет являться plc1, а получателем - plc2. Конечно, можно задать еще достаточно много дополнительных параметров, однако все это можно сделать и позже. Сейчас важно другое - приложение практически создано!



Запустим процесс компиляции проекта, нажав на кнопку **Build Project** и загрузим его в контроллеры через **Download**. После того, как загрузка закончилась, перейдет в режим отладки,



чтобы увидеть изменения переменных во времени. Нажмите на кнопку **Debug**. Теперь Вы можете не только отслеживать состояние переменных, но и вносить в них изменения, наблюдая, как изменится ход программы. Перейдем в Dictionary, развернем список переменных так, чтобы видеть первый и второй ресурс. Первый контроллер изменяет переменную, при этом точно такое же значение появляется и на втором PLC.

Вот Вы и создали простое распределенное приложение в ISaGRAF. Сейчас "контроллеры" работают на соседних столах, однако также успешно они будут работать, будучи разнесенными на огромные расстояния. Эта технология будет востребована инженерами и позволит успешно решать большое количество задач, которые раньше казались просто невозможно сложными.

Изложенный пример можно загрузить с сайта www.isagraf.com.ua.



КОНТАКТЫ:

т. (044) 241-87-39, 492-31-08(09)
e-mail: info@isagraf.com.ua

