

Краткий обзор системы программируемой логики ISaGRAF

Введение

Как только была сформулирована задача программирования логики некоего контроллера, то сразу встал вопрос о соответствующем программном инструментарии для решения этой задачи. Использование средств традиционного программирования, то есть компиляторов с универсальных языков (Си, Паскаль, Фортран, Бейсик и т.п.), дает решение задачи, но требует при этом всеобъемлющих знаний в теории программирования, знаний особенностей конкретной операционной системы и, безусловно, тонкостей аппаратного обеспечения контроллеров, модулей сопряжения с объектом и т.п.

Понятие CASE-инструментов (Computer Aided Software Engineering) бродит в кругах специалистов по системам автоматизации уже давно. Отечественные программисты в соавторстве с технологами пытались и делали системы, включающие идеи и принципы, свойственные понятию CASE, и использовали их для программирования логики контроллеров. Каждая уважающая себя фирма, производящая оборудование для построения систем автоматизации, стремилась сопровождать свою продукцию каким-либо набором программных инструментов, с помощью которых пользователь мог по определенным правилам и соглашениям описывать логику работы контроллера. Эти правила и соглашения оформлялись в виде специальных языков программирования, которые в совокупности с некоторыми элементами человеко-машинного интерфейса (MMI) образовывали так называемый CASE-инструмент.

Все эти системы отличались друг от друга набором поддерживаемых функций, пользовательским интерфейсом и, что самое главное, нестандартными языками программирования логики. Наличие такого "зоопарка" программных средств привязывало пользователя к одной аппаратно-программной платформе, и в этом случае не могло быть речи о переходе на другую. Рано или поздно пользователи должны были "заставить" разработчиков (прежде всего программных средств) учесть их требования по выработке таких программных инструментов, которые позволили бы им не зависеть от особенностей аппаратуры разных производителей.

Настойчивый хор требований привел к тому, что Международная Электротехническая Комиссия (МЭК) в 1992 году выпустила стандарт IEC 1131-3, определяющий пять языков программирования логических контроллеров (PLC). Появление этого стандарта было встречено с большим интересом, и сразу начались работы по созданию программных интегрированных систем, поддерживающих эти языки программирования. В 1990 году была основана французская фирма "CJ International", которая занималась разработкой CASE-инструментов для программирования различных PLC. Эта фирма мгновенно отреагировала на выход стандарта IEC 1131-3 выпуском продукта под названием "ISaGRAF for DOS", а затем и "ISaGRAF for WINDOWS", в которых в полной мере реализовала поддержку всех пяти стандартных языков программирования PLC. ISaGRAF сразу получил признание многих зарубежных фирм. Среди них: ABB, BMW, GENERAL MOTORS, GESPAC, GREEN SPRING, MOTOROLA, PEP MODULAR COMPUTERS, PHOENIX, THEMIS COMPUTER и др. (всего свыше 80 компаний). Данный обзор посвящен, в основном, описанию возможностей и особенностей применения системы программирования ISaGRAF.

Стандарт - Открытость - Результат - Успех

До выхода стандарта на языки программирования PLC производителей контроллеров мало заботил вопрос, какими средствами программировать их "поделки". Каждый работал в рамках своих интересов и возможностей и не очень задумывался о какой-либо совместимости. Сегодня производитель аппаратуры отчетливо понимает, что успех

ему может принести лишь кооперация его продукции со стандартным программным обеспечением. Стандарт на средства программирования даст очевидные преимущества, а именно: получение качественного программного продукта, сопрягаемость на уровне исходных текстов, независимость от типа операционной системы и от конкретной персоны программиста, появление общего языка общения в среде разработчиков прикладного программного обеспечения (ППО) и, наконец, самое важное - значительное сокращение времени разработки ППО и, как следствие, сокращение финансовых затрат на разработку проектов в целом. Пожалуй, единственным "недостатком" такого подхода можно считать некоторые ограничения, являющиеся следствием использования любого стандарта и вынуждающие программиста иногда "наступать на горло собственной песне".

ISaGRAF: спектр возможностей

В ISaGRAF заложена методология структурного программирования, которая дает возможность пользователю описать автоматизируемый процесс в наиболее легкой и понятной форме. Интерфейс с пользователем системы ISaGRAF соответствует международному стандарту GUI (Graphical User Interface), включающему многооконный режим работы, полнографические редакторы, работу с мышью и т.д. Если попытаться сгруппировать основные возможности описываемой интегрированной системы, то их можно представить в следующем виде:

1. Поддержка всех пяти СТАНДАРТНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ PLC (в соответствии со стандартом IEC 1131-3), а именно:

- язык последовательных функциональных схем (Sequential Function Charts, или Grafset), описывающий логику программы на уровне чередующихся процедурных шагов и транзакций (условных переходов);
- язык функциональных блоковых диаграмм (Function Block Diagrams), позволяющий пользователю построить комплексную процедуру, состоящую из различных функциональных библиотечных блоков (арифметических, тригонометрических, блоков управления логикой, PID-регуляторов, блоков списывания некоторых законов управления мультиплексоров и т.д.);
- язык релейных диаграмм, или релейной логики (Ladder Diagrams), используемый для описания логических выражений различного уровня сложности;
- язык структурированного текста (Structured Text), относящийся к классу языков высокого уровня и по мнемонике похожий на Pascal. На основе этого языка можно создавать гибкие процедуры обработки данных;
- язык инструкций (Instruction List), относящийся к классу языков низкого уровня и позволяющий создавать эффективные, оптимизированные процедуры.

Если пользователю недостаточно вышеперечисленных языков, то ISaGRAF для этих целей поддерживает интерфейс к функциям, написанным на языке "ANSI C", но в этом случае требуется компилятор для соответствующей платформы. Для контроллеров фирмы СКБ "Промавтоматика" это компилятор от KEIL Software.

2. ISaGRAF позволяет широко смешивать программы/процедуры, написанные на разных языках, а также вставлять кодовые последовательности из одного языка в коды, написанные на другом языке.

- наличие мощного ОТЛАДЧИКА, позволяющего во время работы прикладной задачи просматривать состояние программного кода, переменных, программ и многое другое;
- поддержка основных ФУНКЦИЙ протокола MODBUS (RTU, SLAVE) для доступа к переменным приложения со стороны SCADA, таких как Intouch, Sitex и других, позволяющих работать с устройствами по протоколу MODBUS;
- наличие дополнительных ИНТЕРАКТИВНЫХ РЕДАКТОРОВ для описания

- переменных, определений и конфигураций ввода/вывода;
- ВСТРОЕННЫЕ СРЕДСТВА КОНТРОЛЯ за внесением изменений в программный код ISaGRAF-приложения и печати отчетов по разработанному проекту с большой степенью детализации, включая печать таблиц перекрестных ссылок для программ и отдельных переменных;
- ПОЛНОЕ ДОКУМЕНТИРОВАНИЕ СИСТЕМЫ разработки и языков программирования (в том числе и на русском языке).

Архитектура ISaGRAF

Условно пакет ISaGRAF можно разделить на две компоненты: систему разработки (ISaGRAF DevSys) и систему исполнения (ISaGRAF Target). КОММУНИКАЦИОННАЯ ЗАДАЧА обеспечивает поддержку процедуры загрузки пользовательского ISaGRAF-приложения со стороны программируемого контроллера, а также доступ к рабочим переменным этого приложения со стороны отладчика системы разработки ISaGRAF. Кроме того, коммуникационная задача реализует так же поддержку протокола MODBUS, что дает возможность доступа к данным на контроллере не только отладчику ISaGRAF, но и любой системе визуализации и управления данными (SCADA), имеющей соответствующий драйвер MODBUS. ДРАЙВЕРЫ УСО (устройства сопряжения с объектом) организуют прозрачный доступ к аппаратуре ввода/вывода и делают этот процесс стандартным для конкретной операционной системы. ФУНКЦИИ ПОЛЬЗОВАТЕЛЯ описывают процедуры и алгоритмы функций, не реализованных в стандартном варианте поставки системы ISaGRAF. СИСТЕМНЫЕ ФУНКЦИИ предназначены для описания специфики конкретной платформы, на которой реализовано данное семейство контроллеров.

И, наконец, ЯДРО ISaGRAF реализует поддержку стандартных языков программирования PLC и стандартного набора функций и функциональных блоков. Процесс разработки ISaGRAF-приложения (задачи пользователя) в общих чертах описывается следующим алгоритмом:

- программирование приложения в рамках интегрированной среды разработки ISaGRAF (ISaGRAF DevSys);
- загрузка приложения из PC в PLC (через RS232) с использованием встроенных средств системы разработки ISaGRAF и коммуникационной задачи со стороны PLC;
- запуск пользовательского приложения под управлением ядра ISaGRAF системы исполнения (ISaGRAF Target);
- запуск интерактивного отладчика со стороны системы разработки ISaGRAF для контроля выполнения пользовательского приложения.

Аппаратно-программные потребности ISaGRAF DevSys ограничиваются стандартной конфигурацией IBM PC-совместимого компьютера:

- PC или Notebook Pentium 100MHz CPU;
- ОЗУ 16 Мбайт;
- 20 МБайт свободного дискового пространства;
- MS-Windows 95 и выше;
- 1 порт RS232 (для связи с контроллером);
- 1 параллельный порт (для ключа программного обеспечения ISaGRAF Workbench).

Основной принцип ISaGRAF: СИНХРОНИЗАЦИЯ

Прикладная задача ISaGRAF работает строго по временным циклам, продолжительность которых определяет разработчик при компиляции задачи. Минимальная продолжительность циклов исполнения прикладной задачи определяется

характеристиками аппаратно-программной платформы (ISaGRAF Target), на которой происходит исполнение задачи. Если параметр синхронизации установить равным нулю, то прикладная задача будет работать по принципу: выполнялась текущая программная последовательность - управление передается следующей без ожидания. Прикладная задача (ISaGRAF-проект) состоит из программ и процедур, которые могут быть, в свою очередь, расположены в трех программных секциях:

1. секция BEGIN (циклическая);
2. секция SEQUENTIAL (последовательная);
3. секция END (циклическая).

Принцип цикличности выполнения прикладной задачи для данного примера работает следующим образом: если величина цикла определена в N мс, это означает, что в начале данного временного цикла выполняются ВСЕ программы/подпрограммы секции BEGIN, в конце цикла - ВСЕ программы/подпрограммы секции END, а в середине цикла выделяется время для выполнения очередных шагов программ/подпрограмм секции SEQUENTIAL. На программы из секций BEGIN и END, как правило, возлагаются задачи по обновлению переменных ввода/вывода. Такая схема работы ISaGRAF-приложения гарантирует пользователю, что в рамках одного временного цикла он будет работать только с одной копией переменных типа INPUT/OUTPUT.

Отладчик

Во все времена ценность и достоинство любой системы разработки программного обеспечения определялась наличием отладочных средств, без которых любая разработка может превратиться в кошмар. Отладчик ISaGRAF предоставляет полный набор возможностей для получения качественного программного продукта (ISaGRAF-приложения), и что немаловажно, все возможности одеты в прекрасный интерфейс. Среди них выделяются следующие:

- поддержка механизма выполнения программ по шагам;
- возможность внесения изменений в код программы во время работы отладчика;
- трассировка рабочих переменных; интерактивная модификация значений переменных;
- останов/запуск отдельных программ, входящих в состав данного приложения ISaGRAF;
- изменение в процессе работы отладчика продолжительности цикла выполнения приложения; эмуляция сигналов, подаваемых на каналы ввода (INPUT) и т.д.

Все эти возможности реализованы в рамках удобного и наглядного графического интерфейса, обеспечивающего своеобразный комфорт в процессе разработки приложения ISaGRAF.

В качестве обобщения

Итак, что же может и что поддерживает ISaGRAF:

- стандартные языки программирования (стандарт IEC 1131-3);
- открытость как инструментальных средств, так и внутренних структур данных прикладной задачи;
- многофункциональный графический отладчик;
- интегрированные средства поддержки протокола MODBUS, а также средств обслуживания последовательных портов RS-232 и RS-485;
- стандартный человеко-машинный интерфейс (MMI);
- возможность связи с любой системой визуализации и управления данными (SCADA-системы) по протоколу MODBUS.

И многое, многое другое. Такой обширный набор инструментов позволяет программировать конфигурации любой сложности.

Все это прекрасно. Но каждый специалист понимает, что за все прелести универсальных стандартных систем приходится чем-то платить. О чём речь? Во-первых, освоение продуктов такого класса, как ISaGRAF, требует некоторых усилий по изучению языков, среды и, конечно, технологии программирования. Хотя все эти компоненты сами по себе несложные, они все-таки требуют серьезного отношения и творческого подхода. Во-вторых, переход на любую новую технологию иногда сопровождается и материальными вложениями, так как пользователю приходится выбирать не просто субъект автоматизации (например, тип контроллера), но и сопровождающий его программный инструментарий. Если выбор сделан правильно - надежный и гарантированный результат обеспечен. Если какой-либо компонент "страдает" серьезными недостатками, то пользователь ощущает глубокое разочарование во всем новом, может махнуть рукой на современные подходы и свернуть на старую проторенную дорогу решения задач "старым дедовским способом". И, в-третьих, что самое важное, пользователю неизбежно придется ломать психологию "получения результата любыми средствами за неопределенный срок" на психологию "получения качественного результата стандартными средствами с минимальными издержками". Если человек сделал выбор в пользу современных технологий и получил при этом качественно другой результат, то заставить его вернуться на старый способ мышления и восприятия задач, достаточно проблематично.

Автор статьи А.Н.ЛЮБАШИН, АО РТСофт, Москва. www.rtsoft.ru