

Все под контролем!

Создание «живых» операторских панелей не требует сколь-нибудь серьезных навыков программирования. С этой задачей сможет справиться любой школьник или студент, знакомый хотя бы Бейсиком

Федоряк Роман, roman.f@rts.ua

Допустим – вы «гуру» в электроприводе и запустили не один преобразователь частоты и не на одном объекте. Тогда Вы наверняка встречались с тем, что вашим приводом (и не только им) в итоге управляет дистанционно оператор, с помощью некоей компьютерной системы, просто тыкая пальцем или мышкой в красивые «живые» картинки на экране, отображающие сам агрегат, меняющиеся цифры и бегущие графики параметров, столбики уровней, стрелки нарисованных приборов и целые мультфильмы с движущимися конвейерами, вращающимися насосами, вентиляторами и бегущей жидкостью.

Хотите научиться самостоятельно делать такие системы? «Не хочу», - скорее всего, ответите Вы, - «Потому что это дорого», или «Потому что у меня аллергия на программирование», «Потому что это не мой бизнес, в конце концов» – возможны любые другие варианты аргументации.

Действительно, для создания подобных систем классическим путем, вам понадобится сотрудничество с еще одним гуру, причем часто капризным – гуру-программистом, а может и не с одним. В самом деле, даже составление спецификации покупных средств автоматизации подобных систем выливается в серьезную задачу, требующую хотя бы эскизного предварительного проектирования.

Для каждой выбранной «железяки» нужно подобрать соответствующий «софт», выбрав из нескольких вариантов, отличающихся по цене в разы. Возможно, придется изучить, что такое «тэг», «точка ввода/вывода», «плавающая лицензия» и что под этим понимает конкретный производитель. Мало того,



каждая недешевая «софтина» может требовать для работы еще одной или нескольких, не менее недешевых программных компонент «общего назначения» от других производителей, например, SQL сервера, или серверной операционной системы.

Большинство производителей возвращают инвестиции в разработку, продавая лицензионное программное обеспечение, защищенное от копирования, имеющее к тому же те или иные ограничения, которые можно снять лишь за дополнительную плату. Наши же реалии приводят к тому, что по рукам ходят «ломанные» и не самые свежие версии, которые неизвестно как работают и по которым нельзя официально проконсультироваться.

В итоге, приводчики предпочитают не связываться даже с простыми задачами, вроде индикации нагрузок и скоростей нескольких двигателей на одном экране, а дистанционное управление преобразователями частоты предпо-

читают делать с помощью физических сигналов и потенциометров. Вот и ходят операторы (либо наладчики) от шкафа к шкафу, записывая на ходу в журнал показания стрелочных приборов или считывая, что показывает индикация на частотниках, и сводят в голове общую картину – правильно ли работает система или нет.

Но ситуация в корне меняется, если производитель принципиально не встраивает в свое ПО никаких ограничений, и более того, распространяет его совершенно бесплатно. Так, у программного обеспечения, разработанного транснациональной корпорацией Delta Electronics нет демонстрационных версий по той простой причине, что все версии являются «боевыми». А самую свежую версию всегда можно загрузить с сайта www.deltaww.com.

Данный материал расскажет читателю, как в этом случае просто и недорого осуществляется стыковка преобразователей частоты Delta Electronics с

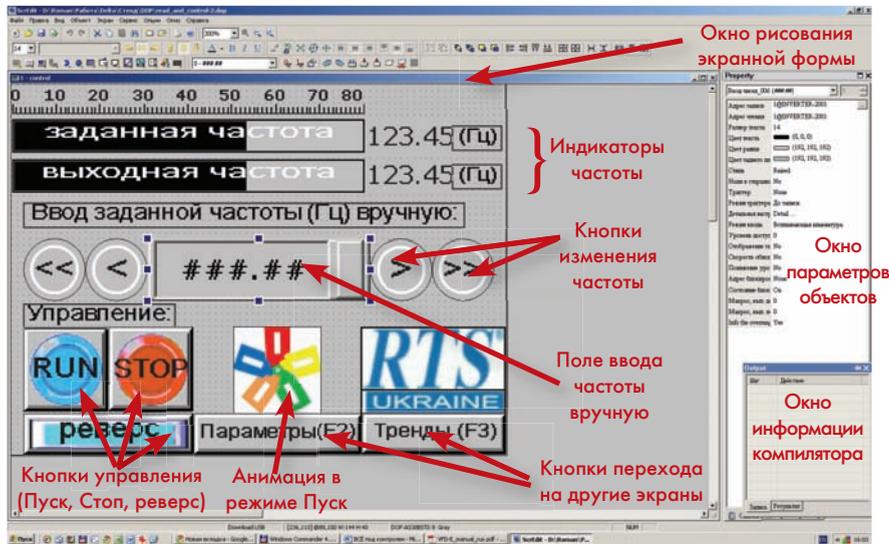


Рис. 1. Графические объекты – кнопки, индикаторы, поля ввода и т.п. – вводятся на экранную форму мышью из банка графических объектов.

сенсорными операторскими панелями того же производителя для получения продвинутого человеко-машинного интерфейса, характерного для современных SCADA систем.

Минимальное программирование, которое может потребоваться для написания макросов, гуру-приводчик вполне может поручить своему сыну – студенту, имеющему представление хотя бы о Бейсике.

Связь по протоколу

Современная приводная техника дошла до такого состояния, что связать различные устройства по известному и открытому протоколу обмена никакими трудностей не вызывает, и это по силам даже лицам, несведущим в тонкостях аппаратной и программной реализации обмена данными. Главное, чтобы все устройства поддерживали единый протокол (говорили на одном языке). Таким единым стандартным протоколом для техники Delta Electronics является Modbus.

Использование данного стандарта позволяет обмениваться данными не только между частотником и операторской панелью, но позволяет включать в единую сеть обмена данными любые устройства – датчики, ПЛК, регуляторы температуры, передавать данные на операторские АРМы. Протоколы Modbus и сети передачи данных на основе стандарта RS-485, где информация передается по витой паре в виде токовых сигналов, подробнейшим образом описаны и здесь мы их обсуждать не будем.

Мы займемся другим – научимся обмениваться данными между преобразователем частоты (а на его месте может быть любое другое устройство с Modbus), и операторской панелью.

Первоначальная настройка

Итак, у нас есть два устройства – преобразователь частоты Delta Electronics (в описываемом примере это VFD007E21T) и операторская панель того же бренда – модель называется DOP-AS38BSTD. У обоих устройств – есть порты RS-485, причем в операторской панели – это клеммная колодка «под винт», в преобразователе частоты – разъем, в который входят вилки типа RJ-45 или RJ-11. Для передачи данных физически используется только одна пара проводов (крайне желательно, чтобы она была витая), длиной – до 1200 метров. Физическое соединение обоих портов витой парой – это все, в чем заключается подготовка для передачи данных.

Будем считать, что преобразователь частоты (ПЧ) – готов к работе, проверен и все необходимые параметры заданы. Единственное, что нужно проверить – это чтобы параметры коммуникации в ПЧ и в операторской панели были заданы единообразно. Для нашего примера возьмем такой случай – Modbus RTU, скорость передачи данных – 9600 бод, используются кадры с длиной сегмента данных 8 бит, 2 стоповых бита, без контроля четности. Эти настройки в ПЧ проводятся путем задания таких параметров, как 09-00=1 – коммуникационный адрес (равный 1), 09-01=1 – соответствует скорости передачи 9600 бит в секунду, и 09-04=3 – протокол Modbus <8,N,2>.

Операторская панель настраивается на эти же параметры обмена данными с помощью своего программного обеспечения, в котором ведется разработка программы. К этому мы и приступаем.

Программирование панели

Операторские панели Delta Electronics программируются с помощью программного пакета «ScreenEditor», который является совершенно бесплатным, русифицированным, и свободно скачивается с сайта Delta Electronics. Связь между операторской панелью и компьютером, на котором установлена программа «ScreenEditor», осуществляется через порт USB, с помощью кабеля, которыми подключаются к компьютеру принтеры, сканеры и прочая периферия. Установка программного обеспечения «ScreenEditor» не заслуживает внимания – это стандартные действия пользователя при установке программ.

Запустив программу «ScreenEditor», создаем новый проект, даем ему имя, устанавливаем протокол коммуникации между операторской панелью и управляемыми устройствами, и приступаем к рисованию нашего операторского интерфейса.

Для начала нужно определиться, какие функции будет выполнять наше приложение. Мы поставили задачу таким образом. На операторской панели должно быть три экрана, переход между которыми осуществляется либо «нарисованными» кнопками, либо физическими – функциональными кнопками самой панели (F1-F3).

На первом экране (рис. 1) мы показываем информацию о заданной частоте и выходной частоте – как в цифровом, так и в графическом виде (полосовой индикатор bargraph). Также мы здесь размещаем поле ввода числового задания на частоту, кнопки быстрого изменения заданной частоты в обе стороны, и кнопки управления – Пуск, Стоп, Реверс. Операторская панель DOP-AS38BSTD оснащена сенсорным экраном, поэтому оператору нужно будет нажимать «нарисованные» кнопки, как будто они настоящие.

Графические объекты – кнопки, индикаторы, поля ввода и т. п., вводятся на экранную форму мышью из банка графических объектов. Разместив объекты на экране, и настроив их свойства (цвет, прозрачность, размер, тип шрифта), остается только настроить их поведение – то есть, заставить конкретные индикаторы читать информацию из указанных регистров преобразователя частоты, а поле ввода заданной частоты и кнопки Пуск, Стоп и Реверс – давать команды в соответствующие регистры частотника.

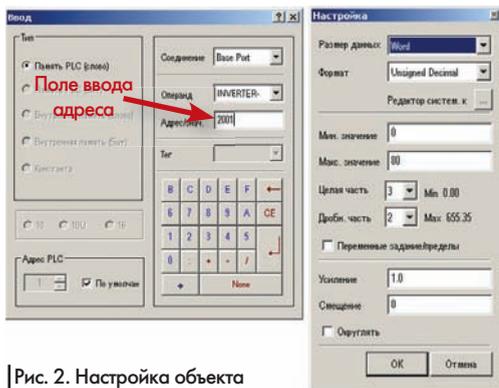


Рис. 2. Настройка объекта

Настройка объектов

Для примера расскажем, каким образом осуществляется настройка поведения поля ручного ввода частоты (рис. 2). Работать оно в реальности должно так – пользователь нажимает на поле ввода частоты, и появляется новое окошко с цифровой клавиатурой, с помощью которой можно ввести новое число. Нажав на кнопку «Ввод» окна цифровой клавиатуры, пользователь подтверждает изменение задания, и данные записываются в соответствующий регистр преобразователя частоты, по сети Modbus.

У графического объекта, который отвечает за ручной ввод частоты, мы должны задать такие свойства – адреса чтения и записи данных, а также установить формат данных – тип переменной, максимум и минимум, разрешение (число знаков после запятой).

Для того, чтобы узнать адрес нужного нам регистра данных (в данном случае – регистра частоты задания), нужно либо обратиться к документации на преобразователь частоты, где все регистры и хранящиеся в них параметры подробнейшим образом расписаны, либо вычислить адрес самостоятельно, пользуясь несложным правилом. В преобразователях Delta Electronics действует следующая система адресации. Адрес состоит из двух шестнадцатеричных слов – одно слово отведено на номер

группы параметров, второе – на номер самого параметра в группе. Если пользователь не знает шестнадцатеричной системы счисления – можно воспользоваться калькулятором с возможностью перевода чисел из одной системы в другую – например, калькулятором из операционной системы Windows. Получается, параметр преобразователя частоты с номером 06-01 (шестая группа, первый параметр), будет иметь адрес регистра 601H в шестнадцатеричной системе, а параметр с номером 10-17 (десятая группа, параметр 17) – адрес регистра 0A11H.

Требуемый нам регистр (заданная частота) имеет адрес 2001H, и этот адрес мы задаем в поле адреса в окне настройки свойств графического поля ввода частоты – см. рисунок.

На том же рисунке рядом, расположено окно настройки формата данных, где задаются размер данных (слово, потому что в случае с преобразователем частоты, мы всегда будем работать со словными регистрами), формат данных (Unsigned Decimal – десятичное число без знака), далее – минимум, максимум, число знаков до и после запятой, коэффициент усиления.

Похожим образом настраиваются и все остальные объекты – кнопки быстрого изменения частоты (расположены слева и справа от поля ввода задания частоты). Также несложно программируются кнопки Пуск, Стоп и Реверс. Нужно всего лишь знать адрес регистров, в который нужно писать команды на запуск, останов и реверс, и сами коды этих команд – это все есть в документации. В нашем случае адрес регистров, в который пишутся команды Пуск, Стоп и Реверс – 2000H, а сами команды – это десятичные числа 2, 1 и 48 соответственно.

Рядом с кнопками Пуск и Стоп расположен анимированный объект – вращающаяся крыльчатка, который показывает текущее состояние привода – запущен

он или остановлен. Анимация достойна отдельного разговора, о ней чуть позже.

Создадим еще два экрана, на одном поместим список некоторых параметров ПЧ, которые можно будет изменять посредством операторской панели, на втором – построим графики изменения заданной и выходной частоты. Особого пояснения процесс создания этих окон не требует – все очень просто. Создаем новое окно, в него перетаскиваем нужные нам объекты (поля ввода данных, текстовые поля, кнопки), настраиваем их поведение и взаимодействие путем выставления нужных свойств в соответствующих окнах программы «ScreenEditor».

Отладка программы

Интересной способностью обладает «ScreenEditor» – он позволяет в режиме симуляции Off-Line (то есть, без подключения к реальному оборудованию), получить на экране компьютера разработчика внешний вид панели, с тем чтобы проверить, как будут выглядеть созданные объекты на ее экране в реальности, и при необходимости – тут же внести корректировки.

Кроме того, есть еще более интересный режим – симуляция On-Line – то есть, работа с подключением компьютера к реальному оборудованию (в нашем примере – к ПЧ). То есть, компьютер на котором стоит «ScreenEditor», имитирует собой поведение операторской панели. Это позволяет окончательно отладить программу, до загрузки ее в реальную операторскую панель. Внешний вид всех трех созданных окон в режиме симуляции On-Line приведен на рис. 3.

После отладки программы в режиме симуляции On-Line, ее остается только загрузить в память операторской панели. Может быть, придется еще подкорректировать цвет и размер объектов, чтобы с ними удобнее было работать на реальном объекте. Время, затраченное на создание и отладку такого приложения, составило в районе 3 часов. И это с уче-



Рис. 3. Режим симуляции On-Line (то есть, с подключением компьютера, на котором стоит «ScreenEditor» и который имитирует собой поведение операторской панели, к реальному оборудованию) позволяет окончательно отладить программу, до загрузки ее в реальную операторскую панель

том того, что пришлось работать с незнакомым программным обеспечением, одновременно изучая документацию.

Создание анимации

Теперь вернемся к одному интересному моменту, который мы оставили «на потом». Это создание анимаций – то есть, движущихся картинок, которые обладают весьма хорошей информативностью и оживляют любую программу. Пользовательские анимации, которые являются еще одной возможностью операторских панелей Delta Electronics – заслуживают внимания еще и потому, что для их создания нужно привлечь к работе встроенный макроязык программирования. Макроязык операторской панели – это мощный инструмент, который превращает операторскую панель в свободно программируемый «вычислитель», способный работать по заданному пользователем алгоритму.

С помощью макроязыка можно получать и обрабатывать данные от различных подключенных к панели источников, выдавать команды на оборудование, видоизменять внешний вид графических объектов. Между тем, он предельно прост и очень похож на Бейсик. Для пояснения некоторых особенностей работы макроязыка, приведем здесь макрос, который осуществляет переключение рисунков, создавая анимированную картинку – вращающуюся крыльчатку на главном экране программы.

В программах для операторской панели Delta Electronics можно писать несколько типов макросов – для нашей задачи «оживления» картинки и создания эффекта движения, мы применим «тактовый макрос», то есть такой, который будет выполняться в фоновом режиме каждый заданный промежуток времени (в нашем случае один раз в 250 миллисекунд). Вот его текст, в левом столбце – номера строк:

1	IFB 1@INVERTER-2101.0 == ON THEN GOTO LABEL 1
2	GOTO LABEL 2
3	LABEL 1
4	IF \$160 >= 9 THEN GOTO LABEL 10
5	\$160 = \$160 + 1
6	GOTO LABEL 2
7	LABEL 10
8	\$160 = 0
9	LABEL 2
10	END

В первой строке макроса, запись «1@INVERTER-2101.0» означает считывание информации из преобразователя частоты Delta (INVERTER), с коммуникационным адресом 1 (1@), из регистра номер 2101H, с выделением нулевого (самого младшего) бита данных- 2101.0. Этот бит содержит единицу (состояние ON), если привод работает, и ноль (состояние OFF), если он остановлен. Поэтому, выделив бит 2101.0 и проверив его состояние (команда битового сравнения IFB в первой строке), мы переходим к той части макроса, которая управляет сменой картинок в графическом объекте «анимация» - GOTO LABEL 1.

Дело в том, что объект «анимация» должен постоянно читать состояние переменной, в которой содержится номер показываемой в данный момент картинки – в нашем случае это переменная из внутренней памяти панели с адресом \$160. Всего картинок 10 штук, поэтому переменная \$160 будет содержать числовые значения от 0 до 9.

Во второй строке макроса, которая выполнится только если бит 2101.0 в состоянии OFF (то есть – привод остановлен), идет передача управления на метку 2 (LABEL 2), после которой стоит команда завершения макроса END – см. строки 9 и 10.

В четвертой строке идет сравнение переменной \$160, отвечающей за хра-

нение номера картинки анимации, с максимальным числом этих картинок – 9. Если это число больше или равно 9, то идет передача управления на метку 10 (LABEL 10 в строке 7).

В пятой строке идет увеличение переменной \$160 на единицу, в шестой строке – переход на конец программы – LABEL 2.

В восьмой строке переменная \$160 (номер картинки) обнуляется.

Данный макрос очень прост, но на самом деле команд в макроязыке достаточно для создания более серьезных программ. В составе встроенных команд макроязыка – основные арифметические и логические операции, операции сравнения и поиска максимума и минимума, операции смещения, переноса и копирования данных в памяти, преобразования данных в различные форматы, команды управления потоком (циклы), вызовы подпрограмм, работа с битовыми операндами. Кроме того, очень интересная группа макрофункций – это команды передачи данных. Эти команды служат цели осуществления обмена данными с устройствами, которые не поддерживают ни протокол Modbus, ни другие зашитые в операторской панели протоколы. При определенных усилиях, можно осуществлять обмен данными с устройствами, работающими по неким специализированным, фирменным протоколам – главное, чтобы был известен формат кадров передачи данных.

Расширенные возможности

С использованием макроязыка операторских панелей, можно создавать достаточно сложные приложения, требующие большого объема вычислений по свободным, задаваемым пользователем алгоритмам. Кроме того, операторские панели Delta Electronics серии DOP можно оснастить некоторыми платами расширения для ввода и вывода дискретных сигналов, что позволит панели давать команды оборудованию напрямую, минуя сетевых «посредников». Для работы с внешними платами



**багатофункціональна
SCADA/Softlogic S³
(SCADA Anywhere)**

На відміну від багатьох інших SCADA:

- ▶ S³ може працювати практично під будь-якою ОС або під декількома одночасно (Windows, QNX, Linux, Solaris...);
- ▶ S³ має потужний інтегрований Softlogic, який є основним, а не допоміжним компонентом згідно стандарту IEC-61131 (ST, FBD, SFC);
- ▶ S³ програмує мережу (класстер) QNX контролерів;
- ▶ S³ вміщує засоби забезпечення додаткової надійності;
- ▶ S³ ліцензується без обмежень по кількості каналів вводу-виводу та змінних.

Спробуй S³ зараз!

Надійність • Зручність • Інтеграція www.s3.com.ua

Таблица 1. Контроллеры и панели, с которыми поддерживают обмен данными операторские панели Delta Electronics.

Бренд	Название/серия контроллера
Delta	Delta Controller Для сервоприводов/ПЧ/регуляторов температуры/ПЛК (984 RTU mode/ASCII mode)
	Delta DVP PLC
Allen Bradley	MicroLogix PLC
	SLC5 PLC
Danfoss	VLT 2800 (FC Protocol)
Facon	Facon PLC
Festo	Festo_PLC
GE Fanuc	90 Series SNP PLC
Hust	Hust CNC Controller
Jetter	Nano Series PLC
	JC Series PLC
Keyence	KV/KZ Series
Koyo	SU/DL Series
	K-Sequence
Lenze	LECOM-A/B Protocol
LG	Master K120S/200S
	Glofa GM6 CNET
	Master-K CNET
LI YAN	LYPLC EX
M2i	M2i Master
	M2i Slave
Matsushita	FP Series
Mirle	FAMA SC
Mitsubishi	FX / FX2N
	A Series/J71UC24
	A2A/A2AS/A2USH A1SH/A3N/A2ASH CPU Port
	Q Series CPU Port
MKS	CT150
Modbus	Modbus (Master) --- 984 RTU/ASCII mode
	Hexadecimal Address (Master) --- RTU/ASCII mode
	nW (Master) --- RTU/ASCII mode
	Modbus (Slave) --- RTU/ASCII mode
Modicon	TSX Micro (Uni-Telway)
	TWIDO
NIKKI DENSO	NCS-FI/FS Series
Omron	C Series
	CJ1/CS1 Series
Siemens	S7 200
	S7-300 (with PC Adaptor)
	S7-300 (without PC Adaptor)
Taian	TP02 PLC
Vigor	M Series
Yokogawa	ACE PLC

ввода/вывода, программное обеспечение «ScreenEditor» оснащено встроенным языком релейно-контактных схем (LD).

Благодаря таким особенностям, как наличие встроенного макроязыка и возможности установки внешних плат расширения, в некоторых слу-

чаях можно обойтись без внешнего контроллера. Для расширения сетевых возможностей панелей Delta Electronics, предусмотрена возможность установки модуля Ethernet, что позволит вести обмен данными также и по протоколу Modbus TCP.

Еще одну возможность операторских панелей Delta Electronics следует отметить. Все они содержат буфер памяти для хранения архива данных – например, истории аварийных сообщений или для построения трендов (графиков) изменения контролируемых переменных. В качестве контролируемых переменных могут выступать как регистры внутренней памяти панели, так и регистры внешнего оборудования, считываемые по сети.

Именно такой возможностью хранения данных из сети в архиве мы и воспользовались, когда строили графики изменения заданной частоты и выходной частоты из ПЧ на экране номер 3 (рис. 3).

В старших моделях семейства DOP объем этого архива данных может достигать 360 МБ, если данные хранятся во внутренней памяти, а с использованием внешнего накопителя данных, подключаемого к панели по USB, – он будет ограничен объемом самого внешнего накопителя. Данные могут записываться в формате CSV, понимаемом программой Excel. Всего можно создавать до 12 различных архивов. Цикл времени записи выборки данных в архив может выбираться в диапазоне 0 ~ 86400000 мс. Кроме того, архивом может управлять не только панель, но и внешнее устройство, подключенное к панели по сети – например, контроллер или даже преобразователь частоты. Инициировать сброс данных в архив можно выставлением какого-либо бита в адресном пространстве самой панели, либо извне.

При работе с преобразователями частоты возможности архивирования данных операторских панелей могут оказаться незаменимыми при отладке системы, поиске неисправностей, «разборе полетов». Преобразователи частоты очень часто лишены часов реального времени, и сообщения об ошибках при этом фиксируют лишь факт события, но не его время, поэтому иногда бывает трудно найти причину сбоя в работе. Но поскольку панели оператора снабжены часами реального времени, то аварийные сообщения, при переносе их в операторскую панель, превращаются в полноценную запись о проблеме, снабженную информацией о времени и текстовым комментарием.

В заключение отметим, что операторские панели Delta Electronics способны общаться не только с частотными преобразователями и другим оборудованием того же производителя, но также снабжены встроенными драйверами для поддержки обмена данными с контроллерами и оборудованием других брендов, неполный список которых приведен в табл. 1. 