

ВЫБОР КОМПИЛЯТОРА НА ОСНОВЕ ОПЫТА РАЗРАБОТКИ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ

П.В.Гайский, А.Н.Логвинчук

Морской гидрофизический
институт НАН Украины
г.Севастополь, ул. Капитанская, 2
E-mail: oaoi@alpha.mhi.iuf.net

Рассматриваются и сравниваются основные детали разработки прикладного программного обеспечения для измерительных приборов в компиляторах и языках программирования таких, как Borland Pascal 7.0, Delphi 6.0, C++ Builder 6.0-7.0 и MS Visual C++ 6.0.

При создании того или иного прикладного программного обеспечения приходится делать выбор наиболее удобного, надежного и оптимального по цене компилятора и языка программирования применительно к решаемой задаче. Поскольку покупка компилятора для внутреннего рынка стран СНГ пока еще не имеет такой важности, то можно кратко сказать, что официальная средняя стоимость пакетов составляет для Borland Delphi 6.0 и C++ Builder 6.0 Enterprise Edition - порядка 3000\$ за каждый в отдельности, а для MS Visual C++ 6.0 - порядка 400\$. Если для потребителя программ для измерительных приборов главные требования лежат в области регистрации, отображения, первичной обработки информации и наглядности интерфейса, то для самого программиста это еще и функциональная надежность и удобство интерфейса компилятора для различных операционных систем и аппаратной базы ПК, качественное описание языка и наличие удобной документации.

Несмотря на бурный рост вычислительной техники для промышленных исполнений в жестких условиях эксплуатации на современной аппаратной базе продолжает широко использоваться ОС MS Dos, а многие вопросы сетевой связи и комбинирования решаются посредством схемотехники, а не программно. В особенности это касается встраиваемых в измерители малогабаритных и малопотребляющих плат класса MicroPC и smartModule, основными задачами которых, как правило, являются регист-

рация информации и минимум автономного управления. Поэтому для таких случаев актуальность программирования в Borland и Turbo Pascal, как наиболее удобных и "испытанных" у нас пакетах, сохраняется. Если говорить о быстродействии арифметических операций с учетом новых 32 и 64-разрядных шин и процессоров, то исполнение их в купе со всеми остальными программными функциями оказывается близко соотносимым как в ОС MS Dos, так и во всех Windows, так как загрузка ресурсов ПК новыми возможностями современных операционных систем всем известно. Что же касается оформления пользовательского интерфейса, то зачастую на этапе первичной обработки и получения измерительных данных его бывает вполне достаточно, тем более в случаях автономных встраиваемых систем, где он вообще не особо нужен.

Однако, со временем необходимость такого первичного программного обеспечения может отпасть. Быстрое развитие способов и объемов хранения программного кода и информации во флеш и др. памяти позволит оттолкнуться от той же MS Dos и "вшивать" в микросхемы довольно мощный и достаточный набор операций. Тогда вся последующая обработка и представление измерительных данных будет заключаться в наглядности отображения, возможности приема-передачи через сеть или новые высокоскоростные порты ввода-вывода, компоновке в ту или иную базу данных и оценке.

Рассмотрим последовательно основные преимущества и недостатки при разработке и функционировании прикладных программ к приборам с использованием указанных выше компиляторов в сравнении. Выбор именно на эти языки программирования пакеты пал в следствие их довольно высокого качества, широкого распространения и удовлетворения требованиям к решению стоящих здесь задач. Хочется отметить, что последующие оценки могут быть довольно субъективным мнением, основанном на определенном жизненном опыте разработок. Под аббревиатурой Borland Pascal будем понимать Pascal для ОС MS Dos.

1. Компилятор и его интерфейс

Borland Pascal - относительно удобен (насколько это вообще возможно для тек-

стового режима) и прост. Компилирует программы текстового и графического (для стандартных EGA/VGA дисплеев – разрешение 640x480 точек, 16 цветов) интерфейса. В то время как, например, компиляторы MS Basic и C того времени графики не имеют. Плохо отслеживает наличие сопроцессора на машине при создании объектного кода, что зачастую приводит к последующим непоправимым ошибкам времени выполнения на операциях с числами типа *double* и *longint*, так часто используемых в точных математических операциях и пакетах под ОС Windows. Никаких подсказок за исключением самого Help-а, который бывает не всегда полным.

Delphi – очень хороший интерфейс, есть почти вся информация об объектах на графической форме, удобные подсказки, хотя некоторая полезная информация затеряна в справочнике слишком далеко. Много встроенных и сторонних дополнительных компонентов, хотя от версии 5 к 6 они не передаются и довольно неэкономно используют ресурсы ПК. При использовании в программе параллельных потоков и рабочей отладке часто закичивается компилятор и система. В общем, оптимальность в использовании ресурсов и устойчивость могли бы быть и лучше. Некоторые нюансы постигаются только методом перебора. Отсутствует автоматическая отмена изменения свойств объектов на формах интерфейса, который и без того довольно сильно подвержен случайному редактированию. Плохо или вообще не отрабатываются некоторые изменения важных свойств объектов (в частности, выравнивание текста) во время выполнения по самой программе.

C++ Builder - Интерфейс данной среды разработки практически не отличается от интерфейса *Delphi*, стабильность работы компилятора намного выше по сравнению с соответствующей версией *Delphi*. При редактировании и разработке возникают все те же небольшие проблемы с отменой изменения свойств объекта. Но в целом данная среда разработки достаточно удобна и приятна в использовании.

MS Visual C++ 6.0 – относительно неудобный интерфейс, в котором отсутствуют подсказки и многие функции, реализованные фирмой Borland. Многого тоже приходится постигать практическим перебором. Нормальной полной панели свойств объек-

тов вообще нет. Поэтому большинство свойств надо задавать программно. При удалении тех или иных объектов с формы отсутствует автоматическое их редактирование в тексте программы. В результате надо это делать опять таки вручную. Вообще, с этим интерфейсом компилятора без солидного и хорошего справочника под рукой (или как минимум MSDN – фирменной системы помощи и примеров) разобраться очень сложно, даже имея некоторый опыт. Возможно, на своих фирменных курсах фирма Microsoft дает что-то дополнительное и особенное, а так самостоятельно придется попотеть.

Стоит отметить неудовлетворительную компиляцию программ в Windows-компиляторах при загруженных в памяти ПК резидентных программах, например, AVP Monitor Касперского. Поэтому от дополнительных сервисов во время программирования и создания объектного кода лучше временно отказаться.

2. Документация и литература

Borland Pascal – вследствие солидного времени присутствия на рынке и большой популярности, как среди разработчиков, так и в учебных заведениях, имеется очень хорошая и доступная литература по данному продукту.

Delphi - довольно много различной литературы по объему и качеству, однако подробное изучение языка и встроенных компонентов осуществляется большей частью из расширенного фирменного Help-а или с помощью специализированных советов, уроков и примеров из электронных версий или интернета (www.bib.com.ua).

C++ Builder - Популярность среды *C++ Builder* среди разработчиков подкрепляется отличной документацией на русском языке, при написании которой используется, как правило, англоязычная справочная информация из Help-а самой среды. Поиск информации в самом Help-е не всегда удобен, но при наличии настойчивости можно найти практически всё.

MS Visual C++ 6.0 – довольно много литературы, но, по большей части, мало полезной для разработки хороших программ. Зачастую – это по-разному переведенная документация (Help), которая и в фирменном виде оставляет желать лучшего и полезна только в качестве справочника

для тех кто все это так или иначе уже знает. Лучшим выходом является использование в качестве обучения примеров из фирменной дополнительно поставляемой системы помощи MSDN, хотя и в ней не все работает должным образом.

3. Ввод-вывод через последовательный порт RS 232 (COM)

Borland Pascal – на данный момент наиболее распространены следующие способы :

- посредством использования прерывания 14H во встраиваемых в программу ассемблерных процедурах;
- посредством управления микросхемой UART, опять таки с помощью ассемблерных команд;
- напрямую через регистры ПК.

Первые два способа работают стабильно и правильно только в системе MS Dos. Третий работает довольно устойчиво и в ОС Windows. Ошибки могут происходить вследствие некоторых драйверов, установленных на ПК и чаще всего прописанных в AUTOEXEC.BAT. Например, драйвер аудио карты фирмы Creative.

Delphi, C++ Builder, MS Visual C++ 6.0 – несмотря на то, что почти для каждого из данных компиляторов существует встроенный компонент работы с последовательным портом, обычными средствами программирования, когда уже разобрался, это достигается довольно просто причем для всех компиляторов одними и теми же функциями. Кстати, предлагаемые в интернете разработки такого плана на тех же стандартных процедурах могут быть неоправданно громоздки. В общем работа с портом осуществляется почти так же, как и с обычным потоковым файлом, который только предварительно должен быть соответственно объявлен и настроен.

Для C++ :

Объявление и открытие :

```
HANDLE hInOutDat = NULL;
COMMTIMEOUTS ctmNew = {0}, ctmOld;
DCB dcbCommPort, dcbCommPort;
LPCTSTR StreamDat;
DWORD dwBytesRead, dwBytesWrite;
CString input_From;
BYTE Info[1];
input_From = T("COM2");
StreamDat = input_From;
hInOutDat = CreateFile(StreamDat, GENERIC_READ |
GENERIC_WRITE, 0, 0, OPEN_EXISTING, 0, 0);
```

Настройка :

```
GetCommTimeouts(hInOutDat, &ctmOld);
ctmNew.ReadIntervalTimeout = 1; // Максимальное
время ожидания каждого байта при чтении в мл.сек.
ctmNew.ReadTotalTimeoutConstant = 100;
ctmNew.ReadTotalTimeoutMultiplier = 0;
ctmNew.WriteTotalTimeoutMultiplier = 0;
ctmNew.WriteTotalTimeoutConstant = 0;
SetCommTimeouts(hInOutDat, &ctmNew);
dcbCommPort.DCBLength = sizeof(DCB);
GetCommState(hInOutDat, &dcbCommPort);
dcbCommPort.BaudRate = 9600; // Частота -
110,300,600,1200,2400,4800,9600,14400,19200,38400,5
6000,57600,115200,128000,256000
dcbCommPort.ByteSize = 8; // Кол-во зн. бит
dcbCommPort.Parity = 'N'; // Паритет - "None",
"Even", "Odd", "Mark", "Space"
dcbCommPort.StopBits = 1; // Стоповые биты -
1,1.5,2
SetCommState(hInOutDat, &dcbCommPort);
Ввод-вывод :
ReadFile(hInOutDat, Info, 1, &dwBytesRead, NULL); //
Чтение из порта 1-го байта в переменную Info
WriteFile(hInOutDat, Info, 1, &dwBytesWrite, NULL); //
Запись в порт 1-го байта из переменной Info
(Если dwBytesRead не равно 0, то операция прошла
успешно)
Закрытие :
SetCommState(hInOutDat, &dcbCommPort);
SetCommTimeouts(hInOutDat, &ctmOld);
CloseHandle(hInOutDat);
```

Для Delphi соответственно :

Объявление и открытие :

```
hInOutDat : THandle;
commtimeouts : TCommTimeouts;
lpOverlappedRead : POverlapped;
DCB : TDCB;
PortName : PChar;
fdwEvtMask,
NBytesRead : DWORD;
ByteIn,
ByteOut : byte;
PortName := 'COM2';
hInOutDat := CreateFile(PortName, GENERIC_READ
or GENERIC_WRITE, 0, 0, OPEN_EXISTING, 0, 0);
Настройка :
GetCommState(hInOutDat, DCB);
DCB.BaudRate := 9600; //
110,300,600,1200,2400,4800,9600,14400,19200,38400,5
6000,57600,115200,128000,256000
DCB.Parity := 0; // (pNone, pOdd, pEven, pMark, pSpace)
DCB.StopBits := 0; // (sb1_0, sb1_5, sb2_0);
DCB.ByteSize := 8; // (sz_4, sz_5, sz_6, sz_7, sz_8);
DCB.Flags := 1;
SetCommState(hInOutDat, DCB);
GetCommMask(hInOutDat, fdwEvtMask);
GetCommTimeouts(hInOutDat, commtimeouts);
commtimeouts.ReadIntervalTimeout := 1;
commtimeouts.ReadTotalTimeoutMultiplier := 0;
commtimeouts.ReadTotalTimeoutConstant := 100;
commtimeouts.WriteTotalTimeoutMultiplier := 0;
commtimeouts.WriteTotalTimeoutConstant := 0;
SetCommTimeouts(hInOutDat, commtimeouts);
Ввод-вывод :
```



```

ReadFile(hInOutDat,ByteIn,1,NBytes-
Read,lpOverlappedRead); // Чтение из порта 1-го
байта
WriteFile(hInOutDat,ByteOut,1,NBytes-
Read,lpOverlappedRead); // Запись в порт 1-го байта
(Если dwBytesRead не равно 0, то операция прошла
успешно)
Закрытие :
SetCommState(hInOutDat, &dcboCommPort);
SetCommTimeouts(hInOutDat, &ctmoOld);
CloseHandle(hInOutDat);

```

Осуществление приема данных при заданных временных настройках можно осуществлять как асинхронно - проверкой информации в буфере порта в нужный момент, так и с помощью создания параллельного потока в программе постоянно опрашивающего открытый для чтения порт и посылающий в основную программу нужную информацию.

4. Графическое отображение результатов измерений

Borland Pascal – один из первых Dos – их компиляторов, начиная с Turbo Pascal, позволяющий писать своими средствами программы для графической среды ПК и сочетать ее с текстовым режимом, включая многооконный режим TurboVision. Максимальные параметры графического режима : разрешение 640x480 точек и 16 цветов. Недостатком является ограниченное количество цветов и поставляемых шрифтов, из которых русифицирован только один самый простой и не качественный. В результате многие надписи в графике приходится полтерно прорисовывать путем линий и точек. Вывод графиков и т.п. производится только самостоятельно программно. Прокрутка больших графических массивов на экране может осуществляться путем копирования-вставки областей экрана через буфер памяти площадью не более 40000 точек.

Delphi, C++ Builder – имеют возможность как самостоятельной графической прорисовки графиков и т.п., так и встроенные компоненты Additional – Chart, ActiveX – Chartfx и VtChart. Довольно мощные и удобные компоненты, но требующие сравнительно большие ресурсы ПК в плане прорисовки на экране дисплея и процессора. Это делает их непригодными при высокочастотной обработке и графическом выводе измерительной информации. Поэтому при прорисовке возможны потеря отдель-

ных точек и даже закливание компьютера. В результате для полей, изолиний и длинных графиков опять таки приходится пользоваться "скроллером" (прокруткой) самостоятельно-построенным через графические средства Canvas, используя оперативную память ПК.

MS Visual C++ 6.0 – приблизительно похожая ситуация как и для Delphi и C++ Builder с учетом своих графических процедур и модулей. В стандартной поставке пакета имеется только один компонент для постройки графиков, на который, как обычно, нет никакой документации.

5. Связь и работа с базами данных (БД)

Borland Pascal – так как автоматическое обновление современных БД с помощью этого языка сейчас, как правило, не предусматривается, то ряды измерений сохраняются в виде отдельных текстовых файлов с разделителями для дальнейшей загрузки в Excel и экспорта в любой необходимый формат.

Delphi, C++ Builder – имеют собственные неплохие пакетные средства поверхностного редактирования и создания любых распространенных современных БД, а также программно-языковые функции управления ими как посредством собственных процедур, так посредством универсальных SQL запросов. Есть возможность создания локальных БД посредством локального сервера. В программах пакета BDE Administrator и Database Desktop периодически случаются сбои, приводящие к сбоям ПК. В самом интерфейсе компилятора присутствует много удобных визуальных компонент для работы с таблицами, связанными графиками и отчетами, как для БД типа Interbase, так и универсальные. Несмотря на рекламируемость БД Interbase, скорость выполнения основных операций сортировки, поиска и добавления средствами Delphi была для идентично больших объемов данных низкой (примерно на порядок) по сравнению с БД Paradox, dBase и Foxpro. Вообще с «неродными» форматами данных Delphi работает значительно медленнее, чем с Paradox-ми. Даже «универсальный» SQL язык для БД здесь имеет отличия, например, от Microsoft-го.

MS Visual C++ 6.0 – имеет встроенные функции работы со всеми теми же широко распространенными форматами БД. Никак

не воспринимает Interbase БД. Вообще «установленные рядом» Interbase и MS SQL сервера друг друга не видят должным образом. Основное рекомендуемое обращение к доступным БД – через SQL запросы. Оно и выполняется быстрее. Естественно для род-

ных и близких к фирме Microsoft баз данных, таких как Foxpro. Здесь для создания, просмотра и использования в глобальном плане БД лучше на машине сразу установить MS SQL Server.

Таблица - Сводная сравнительная таблица оценок

Характеристика	Оценка (по 3 - бальной шкале)			
	Borland Pascal	Borland Delphi	Borland C++ Builder	MS Visual C++
Компилятор (быстродействие и качество)	2	2	3	3
Интерфейс компилятора	1	3	3	1
Документация и литература	3	2	2	1
Работа с последовательным портом	2	2	3	3
Графическое отображение результатов измерений	1	2	2	2
Связь и работа с базами данных (БД)	1	2	2	2

Рекомендуемые пакеты для создания анимации и графических объектов :

Картинки и фотографии – пакет Adobe Photoshop (в особенности при работе с 256 цветной графикой)

Трехмерные объекты и ролики – пакет 3D Studio MAX (как для ОС MS Dos, так и для ОС Windows)

Монтаж звукового сопровождения – пакет Making Waves

Монтаж видео – MGI Video Wave

Выводы

Для использования в качестве компилятора и языка программирования удобней всего все-таки использовать продукты фирмы Borland, тем более что функции SDK, на которых построены все универсальные операции Windows, стандартны и работают в них как и MS Visual C++. Самостоятельное изучение нюансов Visual Studio при отсутствии хорошей литературы занимает много времени и сил. Отечественная информационная и справочная поддержка Delphi и Builder-а гораздо мощнее и доступнее. Вывод графической информации в программе, если не требуется быстрой и сложной математической обработки удобней осуществлять с помощью встроенных компонент, где есть автоматическая перерисовка и др. функции, которые иногда сложно делать самостоятельно. Однако все что универсально и удобно не всегда качественно и лучше всего интерфейс программировать с помощью процедур DirectX SDK или OpenGL, как это делается в играх. Тогда и ресурсы ПК используются оптимально и графика намного красивей.

Для подготовки данных к включению в ту или иную БД, если не задана четкая структура таблиц и связей, ряды данных можно подготавливать как текстовые файлы с разделителями (например «;») для импорта в Excel и дальнейшего экспорта в любой необходимый формат. Если перед разработчиком стоит задача более серьезных операций с БД и есть возможность выбора, то для продуктов фирмы Borland – это формат данных Paradox, а для MS Visual C++ – это Foxpro или MySQL.

В перспективе документация к данным программным пакетам будет по мере освоения русскоязычными пользователями улучшаться. Возможно, появятся собственные компоненты-модули, как, например, быстрое построение графиков в реальном масштабе времени, которые позволят улучшить и облегчить создание интерфейсов и возможности создаваемых программ.

ЛИТЕРАТУРА

1. Грегори, Кэйт. Использование Visual C++ 6. Специальное издание.: Пер. с англ.- М.; СПб.: 1999. – 864 с.
2. Мешков А.В., Тихомиров Ю.В. Visual C++ и MFC. – 2-е изд.- СПб.: БХВ-Санкт-Петербург, 1999. – 1040 с.
3. Культин Н.Б. Delphi 6. Программирование на Object Pascal. – СПб.: БХВ-Петербург, 2001. – 528 с.
4. Архангельский А.Я. Программирование в C++ Builder 5. – М.: ЗАО «Бином», 2001. – 1152 с.