

ЗНАЧЕНИЕ ГРАФИЧЕСКОГО ПРОГРАММИРОВАНИЯ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

Недорезов Д. А.

**Открытое акционерное общество
Информационные спутниковые системы
имени академика М. Ф. Решетнева**

Во все времена в мире в процессе общения и взаимодействия людей ведущая роль отводилась невербальному общению - языку жестов, движений и образов, без использования словесной речи и письменности. Мы очень часто обращаемся к этому способу передачи информации, общаясь с друзьями, родственниками, деловыми партнерами, сослуживцами. Он во многом определяет как реакцию окружающих, так и их отношение к нам. Стоит только сознательно отнестись к этим безмолвным сигналам, которые мы одновременно и подаем, и принимаем, как мы тут же откроем для себя возможность более эффективного их использования. Проще говоря, самые разные внешние признаки, такие как выражение лица или направление взгляда, помогают нам догадаться о чувствах других людей и об их намерениях по отношению к вам. По мнению психологов, до восьмидесяти процентов информации человек получает по визуальным каналам восприятия. Наряду с речевым восприятием, между людьми, постоянно передаются сообщения, которые они принимают и на которые они реагируют, прежде чем сознание подвергнет их рациональному анализу. Каждый, отдельно взятый человек, как и все человечество в целом в начале своего существования передавал сообщения миру не словами и не буквами, а по средствам мимики и жестов. Язык образов инстинктивно более понятен и близок нам, быстрее и проще осваивается, т.е. играет ведущую роль в образовательном процессе. Также верно и утверждение, что первые нанесенные человечеством письмена представляли собой отражение не букв и звуков, а образов. Более того, в письменности некоторых народов образ не потерял своих ведущих позиций, речь идет о тех этносах, которые используют иероглифическую письменность – японцы, китайцы и т.д.

Рассмотрим историю развития программирования электронных вычислительных машин (ЭВМ). Первые программы для ЭВМ заключались в установке ключевых переключателей на передней панели вычислительного устройства. Очевидно, таким способом можно было составить только небольшие алгоритмы. С развитием компьютерной техники появились языки программирования, но в начале своего развития программирование основывалось на аппаратно ориентированных языках низкого уровня, прекрасно понимаемых машинами, но крайне сложными и неудобными для человека. Затем уровень начал подниматься, появились более автоматизированные системы разработки для конкретных аппаратных средств (ассемблеры), но даже работа с ассемблерами достаточно сложна и требует специальной подготовки, поэтому появились языки высокого уровня. В 1954 году был создан первый язык высокого уровня — ФОРТРАН (англ. *FORT*RAN - *FOR*mula *TRAN*slator). Языки высокого уровня имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы. Эти языки более удобны для человека, с помощью них, можно писать программы до нескольких тысяч строк длиной. Однако легко понимаемый в коротких программах, этот язык становился нечитаемым и трудно управляемым, когда дело касалось больших программ. Решение этой проблемы пришло после изобретения языков структурного программирования (англ. *structured programming language*), таких как Алгол(1958), Паскаль(1970), Си(1972). Суть такого подхода заключается

в возможности разбиения программы на составляющие элементы. Хотя структурное программирование, при его использовании, дало выдающиеся результаты, даже оно оказывалось несостоятельным тогда, когда программа достигала определенной длины. Для того чтобы написать более сложную (и длинную) программу, нужен был новый подход к программированию. И в конце 1970-х и начале 1980-х годов были разработаны принципы объектно-ориентированного программирования (ООП) сочетающие лучшие принципы структурного программирования с новыми мощными концепциями, базовые из которых называются инкапсуляцией, полиморфизмом и наследованием. Примерами объектно-ориентированных языков являются C++, Java и др. Далее произошел качественный скачок – появление визуализированных средств разработки программного обеспечения (C++ Builder, Delphi), которые, содержат, помимо, множества стандартных функций и библиотек, средства визуализации. Они позволили значительно облегчить и оптимизировать работу программистов, но не позволяли программировать людям далеким от этого, например инженерам-конструкторам. Такую возможность предоставило появление революционной технологии графического программирования. Простота освоения данной технологии обусловлена, прежде всего, интуитивностью. Все люди на земле, независимо от того какими вербальными языками они владеют, понимают язык образов, образ – это основа нашего восприятия. В графических языках программирования все объекты, используемые для разработки представлены в виде образов, поэтому процесс создания программного обеспечения (ПО) не требует знаний в области текстовых языков. Как мы видим из вышеизложенного наблюдается тенденция на развитие систем в направлении простоты и скорости разработки, т.е. удобства для программиста и опять же история возвращает нас к оперированию образами, вместо оперирования вербальными единицами, т.к. образ проще для человеческого восприятия, конечно же стоит отметить и то, что для изображения на бумаге удобнее использовать буквы, ввиду простоты их написания.

Наиболее ярким представителем графических систем программирования, на сегодняшний день, является система LabVIEW (Laboratory Virtual Instrument Engineering Workbench) международной корпорации National Instruments. Данная графическая система не уступает по функциональным характеристикам другим системам программирования высокого уровня, а в части быстроты разработки кодов и простоты освоения не имеет себе равных. В отличие от текстовых языков, где программы составляются в виде строк текста, в LabVIEW программы создаются в виде графических диаграмм (блок диаграмм см. рис. 1), подобных обычным блок-схемам и напоминающим электрические схемы, что делает эту систему дружелюбной для инженеров разработчиков радиоэлектронной аппаратуры. Автоматизированные информационные системы, разработанные в LabVIEW называют виртуальными приборами, а интерфейс конечного пользователя, который отображается на мониторе ЭВМ и через который управляется система называется лицевой панелью виртуального прибора (см. рис.2). LabVIEW содержит множество объектов разработанных специально для работы с радиоэлектронной аппаратурой (РЭА), это избавляет от затрат на их создание. Среда позволяет разрабатывать прикладное программное обеспечение для организации взаимодействия с измерительной и управляющей аппаратурой, сбора, обработки и отображения информации и результатов расчетов, а также моделирования, как отдельных объектов, так и автоматизированных систем в целом. В состав LabVIEW входят библиотеки управления различными аппаратными средствами и интерфейсами, такими как PCI, cPCI/PXI, VXI, GPIB (КОП), VISA. Программные продукты, созданные с использованием LabVIEW, могут быть дополнены фрагментами, разработанными на других языках программирования. Кроме того, подавляющее число разработчиков серийной контрольно-измерительной аппаратуры снабжают свою продукцию драйверами для LabVIEW, что избавляет разработчика ПО от низкоуровневого программирования. Таким образом,

LabVIEW позволяет разрабатывать практически любые приложения, взаимодействующие с любыми видами аппаратных средств, поддерживаемых операционной системой компьютера и идеально подходит для автоматизации работы с РЭА.

Среда LabView спроектирована максимально дружелюбно для инженеров-разработчиков РЭА, т.е. для людей, изначально, далеких от программирования. Это делает ее наиболее подходящей для начала обучения их программированию с возможным последующим совершенствованием и изучением других, более не привычных систем программирования, таких как C++, Pascal и т.п. На примере LabView у этих учеников наглядно и понятно сформируются в сознании базовые универсальные понятия программирования, такие как процедуры, функции, циклы, условия, константы, переменные, типы данных и т.п. И сформируются эти понятия более конкретно, в виде картинок и физических наглядных связей, форм взаимодействия элементов, в отличие от абстрактных понятий и вербализованного программирования.

Подобно LabView создаются и будут создаваться другие системы, включающие элементы графического программирования, такие как, AudioMulch - для создания музыки, основанная на потоке звукового сигнала, Miracle C Compiler - среда графического программирования, позволяющая формировать матрицу отношений компонентов платформы. Среда, адаптированная к другим предметным областям, и просто усваиваемые специалистами различных профилей. Организуется процесс обучения программированию подобный процессу обучения детей языкам устной речи и письменности – вначале изучаются образы, затем вербальные значения, ассоциируемые с этими базовыми образами.

Подобный подход обеспечит появление качественно новых кадровых специалистов, которые будут объединять в себе как минимум двух профессионалов: специалиста в предметной области и специалиста по автоматизации данной предметной области. Что решит многие проблемы и позволит максимально оптимизировать производственный процесс. Исчезнут проблемы непонимания различными профессионалами друг друга. Ведь на практике бывает очень трудно договориться, например, гуманитария, работающему с обширной базой данных, связанной с его работой или экономисту, хорошо понимающему как управлять предприятием, и слабо понимающему языки программирования, стоящими с одной стороны. И с другой стороны программистом, имеющим технический склад ума и отлично знающим языки программирования, но слабо понимающим предметную область автоматизации.

Таким образом, мы можем заключить, что технологии графического программирования могут стать неопределимым подспорьем в процессе обучения специалистов в области разработки автоматизированных информационных систем. И следует уделять более пристальное внимание вопросу их изучения и внедрения в образовательный процесс.

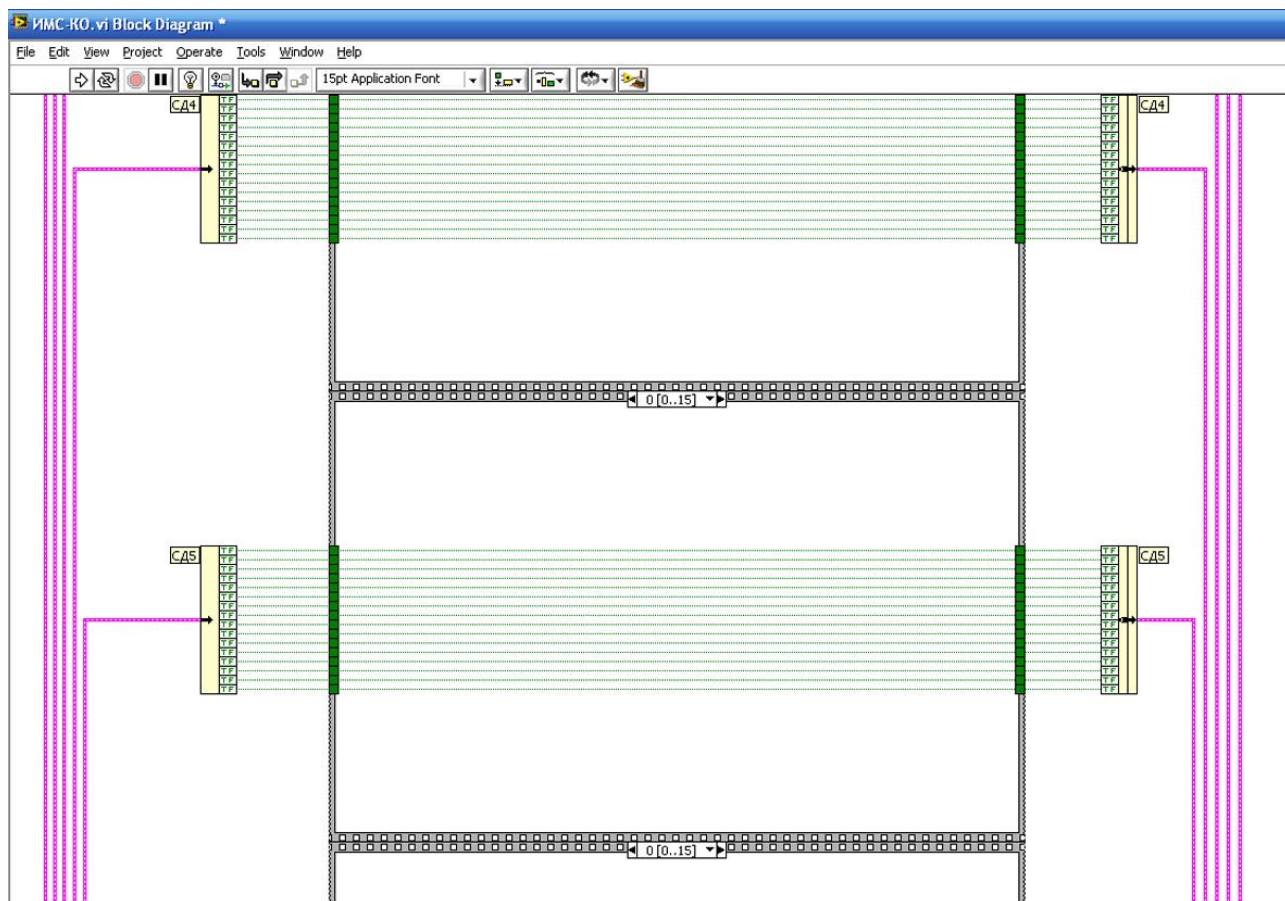


Рисунок 1 - Блок-диаграмма виртуального прибора разработанного в LabView.

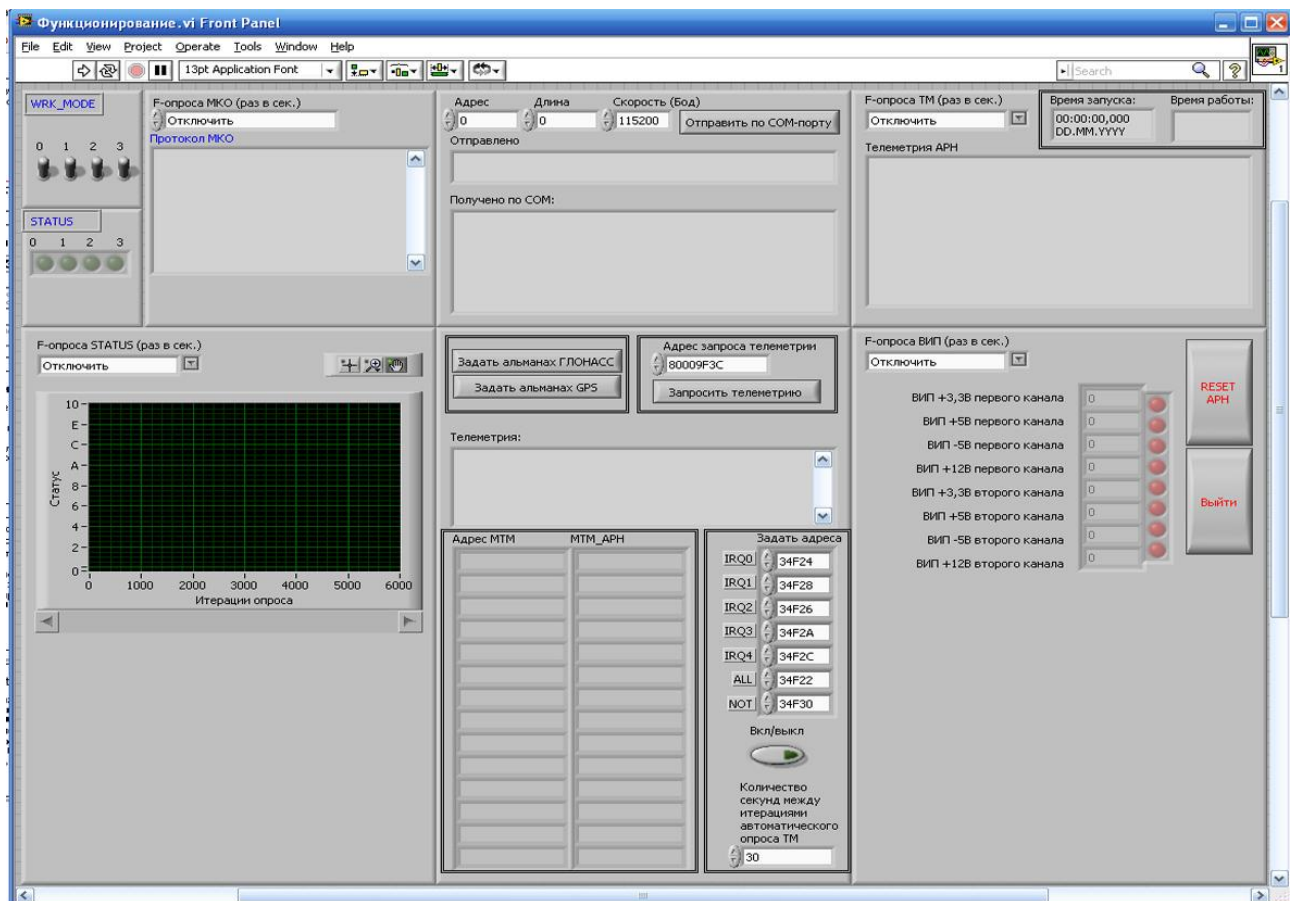


Рисунок 2 - Лицевая панель виртуального прибора разработанного в LabView.