

Построение автоматизированных испытательных стендов с помощью системы RealLab! и MS Excel

Виктор Денисенко,
к.т.н.

Victor@RLDA.ru

Введение

Разработка и производство качественной электронной аппаратуры требует проведения испытаний на соответствие аппаратуры и ее компонентов требованиям технических условий, на стойкость к воздействию температуры, вибрации, ударов, влаги, давления, радиации, света и т. п. С появлением IBM PC-совместимых компьютеров построение таких стендов стало достаточно простым делом. В настоящей статье описывается новая отечественная разработка RealLab!, которая позволяет создавать испытательные стенды, не имея опыта программирования или знания специализированных аппаратных средств. Достаточно только иметь некоторые навыки работы с программой Microsoft Excel, которая установлена практически на каждом компьютере. Особенностью данного подхода является возможность оперативно изменять управляющую программу, обращаться к устройству ввода-вывода аналоговых сигналов и делать вычисления прямо в ячейках электронной таблицы. Кроме того, применение IBM-совместимого компьютера и стандартных программных средств обеспечивают достаточно высокую надежность программного и аппаратного обеспечения в силу колоссального тиража и объема тестирования этих средств.

В принципе язык Visual Basic for Application (VBA), поддерживаемый MS Excel, позволяет решить практически любую задачу, стоящую перед проектировщиком испытательного стенда, однако мы рассмотрим только те из них, которые могут быть решены в короткий срок рядовым пользователем персонального компьютера, не имеющим навыков программирования.

Подключение аппаратуры к MS Excel

MS Excel поддерживает технологию динамического обмена данными (DDE), динамически связываемые библиотеки (DLL) и модель многокомпонентных объектов (COM). Все эти средства могут быть использованы для связи между программой, выполняющей ввод-вывод, и MS Excel. Особо следует отметить COM-технологию, которая является базой для построения OPC клиента и сервера и которой Microsoft в последнее время придает особенно большое значение. Данная тема требует отдельного рассмотрения; мы же приведем только пример подключения к MS Excel аппаратных средств серии RealLab! фирмы RLDA с помощью COM технологии.

Прежде всего, COM-объект требует своей регистрации в системном реестре Windows. Для этого используется команда -RegServer, для выполнения которой в командной строке следует набрать, например:

```
C:\DAQ\COM_RL.exe -RegServer
```

где COM_RL.exe — название COM-объекта, C:\DAQ\ — путь к нему. После этого COM-объект становится «виден» с помощью пункта меню Tools/References в VBA, который запускается выбором пункта Сервис/Макрос/Редактор Visual Basic на панели MS Excel. Найдя нужный объект в открывшемся списке (в рассматриваемом примере он называется COM_RL 1.0 Type Library), следует отметить его флажком и закрыть окно. После этого все функции работы с системой ввода-вывода, принадлежащие COM-объекту, становятся доступными в редакторе VBA, и их можно использовать для дальнейшей работы. Ниже мы будем считать, что данные из внешнего устройства приняты и хранятся в переменной «Data», а подготовленные для вывода данные хранятся в переменной «DataOut».

Устройство ввода-вывода аналоговых сигналов RealLab! (рис. 1) подключается к компьютеру через интерфейс RS-232, который может иметь гальваническую развязку. Длину кабеля для подключения последовательного интерфейса можно увеличить до 1,5 км с помощью преобразователей RS-232 в интерфейс RS-422 или «токовая петля». В комплект системы RealLab! входят также устройства сопряжения с разнообразными датчиками, что позволяет конфигурировать аппаратную часть системы в соответствии с требованиями заказчика.

Создание органов управления

Примеры типовых органов управления, функционирующих в MS Excel, представлены на рис. 2. Эти рисунки, выполненные в растровом формате и вставленные в Excel. При нажатии мышью на пра-



Рис. 1. Устройство ввода-вывода аналоговых сигналов RealLab!

вую стрелку переключателя пределов измерения (рис. 2, а) белая точка, указывающая предел измерения, перемещается на один шаг по часовой стрелке; при нажатии на левую кнопку точка движется в обратном направлении. В заранее заданной ячейке (например, А3) появляется число, указывающее номер позиции переключателя. Этот номер можно использовать, например, для масштабирования данных, принятых от системы их сбора.

Точка, указывающая положение переключателя на рис. 2, а, является овалом, нарисованным средствами Excel. Она не может быть рисунком в растровом формате, поскольку VBA, входящий в состав Excel, не позволяет выполнять поворот растровых рисунков.

Важно отметить, что код, управляющий движением рассматриваемых ниже органов управления и контроля, можно не писать, а включить режим записи макроса, и в это время произвести нужные операции над создаваемым объектом. В результате Excel автоматически создаст код, выполняющий то, что вы делали вручную. Это избавляет вас от изучения языка VBA. Конечно, желательно понимать смысл того, что создал вместо вас мастер макросов MS Excel, но для этого достаточно знать английский язык, который помогает понять смысл отдельных команд записанного макроса.

Макрос, перемещающий точку по часовой стрелке (рис. 2, а), выглядит следующим образом:

```
GaugePosition = Sheets(«Gauge»).Range(«A3»)
Sheets(«Gauge»).Shapes(«Point»).IncrementRotation 22
GaugePosition = GaugePosition + 1
Sheets(«Gauge»).Range(«A3»).Value = GaugePosition
```

Здесь первая строка считывает начальную позицию переключателя, которая хранится в ячейке А3 на листе Gauge. Вторая строка выполняет поворот точки Point относительно центра вращения на угол 22°. Для создания центра вращения в нужном месте следует создать вторую точку симметрично относительно центра ручки, сделать ее невидимой и сгруппировать с видимой точкой. Такая группа в нашем примере названа Point. Третья строка представляет собой стандартный счетчик с приращением 1, четвертая строка записывает значение счетчика в ячейку А3. Переменную GaugePosition можно использовать в дальнейшем для масштабирования принятых данных.

Слайдер (рис. 2, б) является стандартным элементом Excel, мы добавили к нему лишь шкалу. Позиция указателя записывается в ячейку, которую вы выбираете сами. Кнопки СТОП и ПУСК являются стандартными кнопками MS Excel, изменен только их цвет. Оригинальная кнопка (рис. 2, в) является растровым рисунком, а зеленый светодиод на ней — векторным рисунком, выполненным средствами Excel. При нажатии кнопки светодиод гаснет (его цвет изменяется на серый). При этом в ячейку А6 записывается значение False; при повторном нажатии светодиод загорается, и в ячейку записывается значение True:

```
If Worksheets(«Gauge»).Range(«A6»).FormulaR1C1 = «=TRUE()» Then
Worksheets(«Gauge»).Range(«A6»).FormulaR1C1 = «=FALSE()»
Worksheets(«Gauge»).Shapes(«Lights»).Fill.Visible = msoTrue
```

```
Else
Worksheets(«Gauge»).Range(«A6») = «=TRUE()»
Worksheets(«Gauge»).Shapes(«Light»).Fill.Visible = msoFalse
End If
```

Третья строчка кода делает светодиод видимым (Fill.Visible = msoTrue), а шестая — гасит его.

MS Excel позволяет «закрепить» каждый элемент на листе, защитить от случайного удаления его и весь рабочий лист. Данные можно вводить непосредственно в ячейки рабочего листа Excel, в текстовые поля или в специальные элементы управления, поставляемые в составе MS Excel.

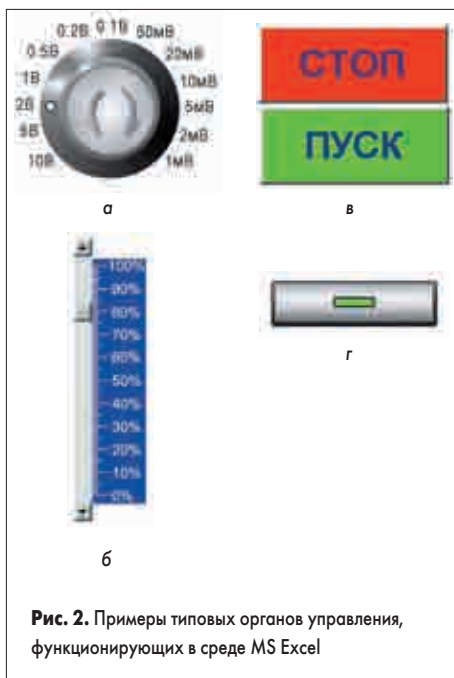


Рис. 2. Примеры типовых органов управления, функционирующих в среде MS Excel

Создание элементов индикации

Элементы индикации строятся аналогично. Неподвижная часть создается в любом графическом редакторе или просто вводится сканированное изображение фотографии реального объекта. Подвижная часть может быть создана либо в растровом формате, либо в векторном — средствами самого Excel. Важно отметить, что набор функций для управления этими типами объектов различен. Например, растровый формат невозможно повернуть на заданный угол, а векторный формат невозможно масштабировать относительно первоначального размера. Поэтому стрелка выполнена в векторном формате (рис. 2, а), а красный столбик указателя — в растровом (рис. 3, б), что позволило масштабировать его относительно исходного размера.

Код, поворачивающий стрелку на угол AngleData, предельно прост:

```
ActiveSheet.Shapes(«Arrow»).Select
Selection.ShapeRange.Rotation = AngleData
```

Естественно, величина угла должна быть связана с принятыми данными Data, например, соотношением:

$$Data = DataMax / 340$$

где 340 — максимальный угол поворота стрелки, DataMax — предельное значение шкалы измерения. Угол задается относительно нулевого значения, поэтому возвращать

стрелку в первоначальное положение не требуется.

Вывод числового значения Data или текстовой строки на ЖКИ-индикатор (рис. 3, б) выполняется следующим образом:

```
ActiveSheet.Shapes(«Text»).Select
Selection.Characters.Text = Data
```

Первая строка выбирает элемент «Текстовое поле», который вставлен поверх растрового изображения ЖКИ-индикатора и сделан прозрачным; вторая строка выводит число. Важно отметить, что шрифт, его размер и цвет, а также цвет текстового поля задаются обычными средствами Excel непосредственно на рабочем листе и в текст макроса их можно не заносить.

Высота красного столбика (рис. 2, б) задается также всего лишь двумя строками:

```
ActiveSheet.Shapes(«Column»).Select
Selection.ShapeRange.ScaleHeight Data, msoTrue, msoScaleFromBottomRight
```

Аналогичный код можно использовать для изменения «уровня воды» в баке (рис. 3, д) на величину Data:

```
Worksheets(«Gauge»).Shapes(«Water»).IncrementTop Data
```

Этот код перемещает растровое изображение воды вниз на величину Data.

Вращение лопастей насоса (рис. 3, з) выполняет код

```
While Sheets(«Gauge»).Range(«A5»).Value = 1
Time = Timer
Sheets(«Gauge»).Shapes(«Paddle»).IncrementRotation 40
While Timer < Time + 0.1
DoEvents
Wend
Wend
```

Здесь рисунок насоса является растровым, а его лопасти выполнены средствами Excel, чтобы сделать доступным метод IncrementRotation поворота на заданный угол (40°); Paddle — имя рисунка лопастей. Команда DoEvents на заданное время (в нашем примере 0,1 с) передает управление Windows, чтобы Excel мог выполнить другие макросы, пока лопасти вращаются.

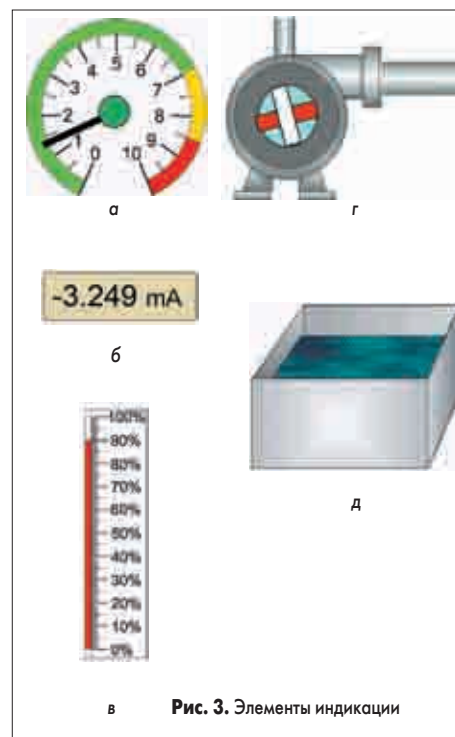


Рис. 3. Элементы индикации

Заключение

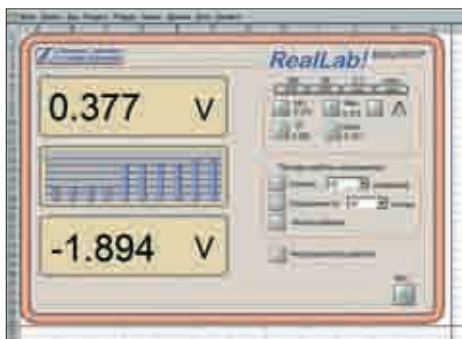


Рис. 4. Виртуальный вольтметр с автоматическим выбором пределов измерения

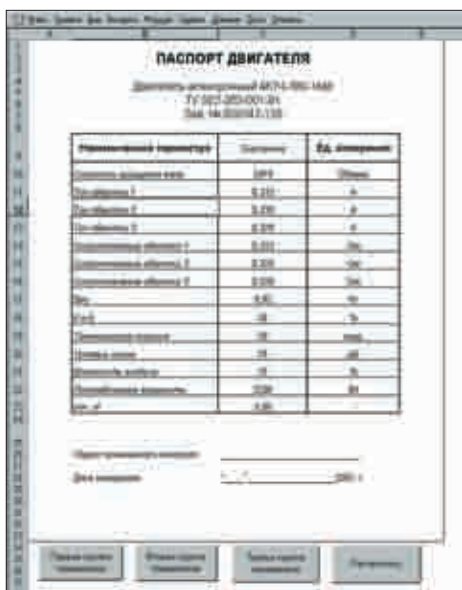


Рис. 5. Паспорт двигателя, заполняемый автоматически данными, полученными из измерительного стенда

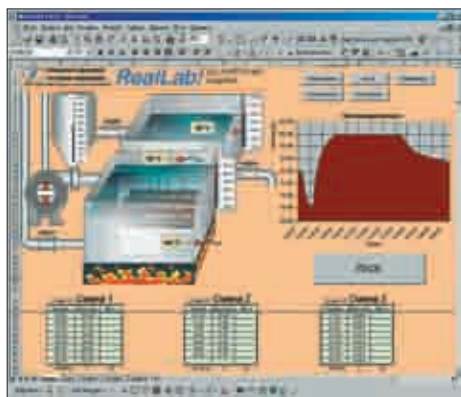


Рис. 6. Программа управления технологическим процессом изготовления пенополистирольных изделий

усреднения измеренных данных и процедура автокалибровки написаны средствами VBA. Программа позволяет также хранить заданное количество измеренных данных для их последующего анализа.

Паспорт двигателя (рис. 5) полностью создан средствами, входящими в MS Excel. Данные поступают из системы сбора данных прямо в ячейки таблицы и могут быть распечатаны и сохранены в файле, причем эти операции выполняются средствами самого Excel.

Система RealLab! позволяет также создавать несложные системы управления технологическими процессами. В качестве примера на рис. 6 показана программа автоматизации техпроцесса изготовления пенополистирольных изделий. ЖКИ-индикаторы отображают температуру термопар, вертикальные шкалы — уровень жидкости в баках, а график в динамике показывает ход технологического процесса. Температурный режим работы системы задается с помощью алгоритма, описанного на языке VBA.

Особенности применения MS Excel

Отметим некоторые общие особенности создания программ автоматизации с помощью MS Excel. После отладки программы ее следует защитить от случайной порчи исходного текста. Для этого используется пункт меню Сервис/Защита. Перед защитой в свойствах каждого элемента следует указать, должен ли он быть защищен после защиты всего листа. Если некоторые элементы должны изменяться программно, сначала нужно снять их защиту, а после модификации элемента вновь защитить его. Однако есть более простой способ: поверх изменяемого элемента можно положить «стекло» — прозрачный растровый рисунок — и защитить его. Тогда лежащие под «стеклом» объекты будут недоступны пользователю, а программа сможет их модифицировать.

При передаче такой программы заказчику ее текст можно скрыть с помощью пункта меню VBA Tools/VBA Project Properties и назначить пароль, который не позволит пользоваться исходным текстом программы без санкции поставщика.

Описанным способом были написаны программы виртуального вольтметра (рис. 4), генерации паспорта двигателя (рис. 5) и управления технологическим процессом вспенивания пенополистирола (рис. 6). Автокалибровка выполнена с помощью VBA, вычисление параметров функции распределения ошибок — с помощью функций MS Excel. Для отображения графиков вообще не было написано ни строчки кода. Данные, отображаемые на графике, берутся из ячеек MS Excel, куда они предварительно записываются кодом

```
Sheets(GraphName).SetSourceData Source:= _
Sheets(Data).Range(Sheets(Data).Cells(13, j + 1), Sheets(Data).Cells(iMax, j + 1)), _
PlotBy:=xlColumns
```

Здесь j — переменная цикла, в котором подлежащие данные записываются в нужные ячейки рабочего листа Data. При обновлении данных в ячейках Excel автоматически обновляет график, поэтому график оказывается «живым», он обновляется по мере поступления данных. Числа, отображающие значение математического ожидания МО, среднеквадратического, минимального и максимального значений на рис. 4, также изменяются в динамике. Частота обновления задается пользователем в ячейке MS Excel. Функции