

Н.Гусев, А.Каракулов, Г.Родионов

## **Delfino™ и студия MexBIOS ускоряют разработку промышленных систем**

### *Аннотация*

*В статье рассматривается концепция и реализация высокоуровневой системы автоматизированного проектирования (САПР) для разработки программ микроконтроллерных систем управления. Основная область применения разработанного продукта – системы управления электродвигателями и устройствами автоматизации. Система позволяет подключить к созданию «интеллекта» встраиваемой системы разнопрофильных специалистов, организовать командную работу и спроектировать систему с учетом особенностей микроконтроллера, объекта управления, условий применения. Работа системы MexBIOS демонстрируется на примере создания программы управления электроприводом для модуля TE-TMS320F28335 компании Терраэлектроника.*

### **1. Традиционное программирование микроконтроллерных систем**

В настоящее время для программирования микроконтроллерных систем управления используются в основном языки высокого уровня, часто Си. Традиционные системы программирования, такие как IAR Workbench, Keil, Code Sourcery и другие имеют развитые средства поддержки работы программиста: редакторы текста с поддержкой синтаксиса, оптимизирующие компиляторы, развитые библиотеки, отладчики с возможностями моделирования и трассировки.

Эффективность работы может быть существенно увеличена, если создать систему проектирования, «дружественную» для специалистов предметной области, знающих особенности работы устройств со специфичными объектами управления. Таким образом, убрать превалирование микроконтроллерной специфики и особенностей сред программирования над оптимальными алгоритмами обслуживаемых процессов.

Решение указанных проблем возможно при создании программных надстроек над стандартными системами программирования, которые позволят проектировщикам устройств абстрагироваться от искусства программирования микроконтроллеров и сосредоточиться на оптимальной реализации задач управления.

Компания «Мехатроника-Про» на основе опыта собственных разработок создала программный продукт САПР MexBIOS (далее Система), являющийся дополнением к существующим системам программирования и позволяющий выполнить практически полный цикл проектирования встроеного программного обеспечения микроконтроллеров – от предварительного моделирования разрабатываемой системы и пробных запусков на макетах, и до работ по сопровождению производства, внесению изменений в выпускаемую продукцию. Основная область применения разработанного продукта – системы управления электродвигателями и устройствами автоматизации, однако примененный подход универсален и для других направлений. Компания является участником Texas Instruments Developer Network и одним из немногих поставщиков средств разработки для микропроцессоров производства TI.

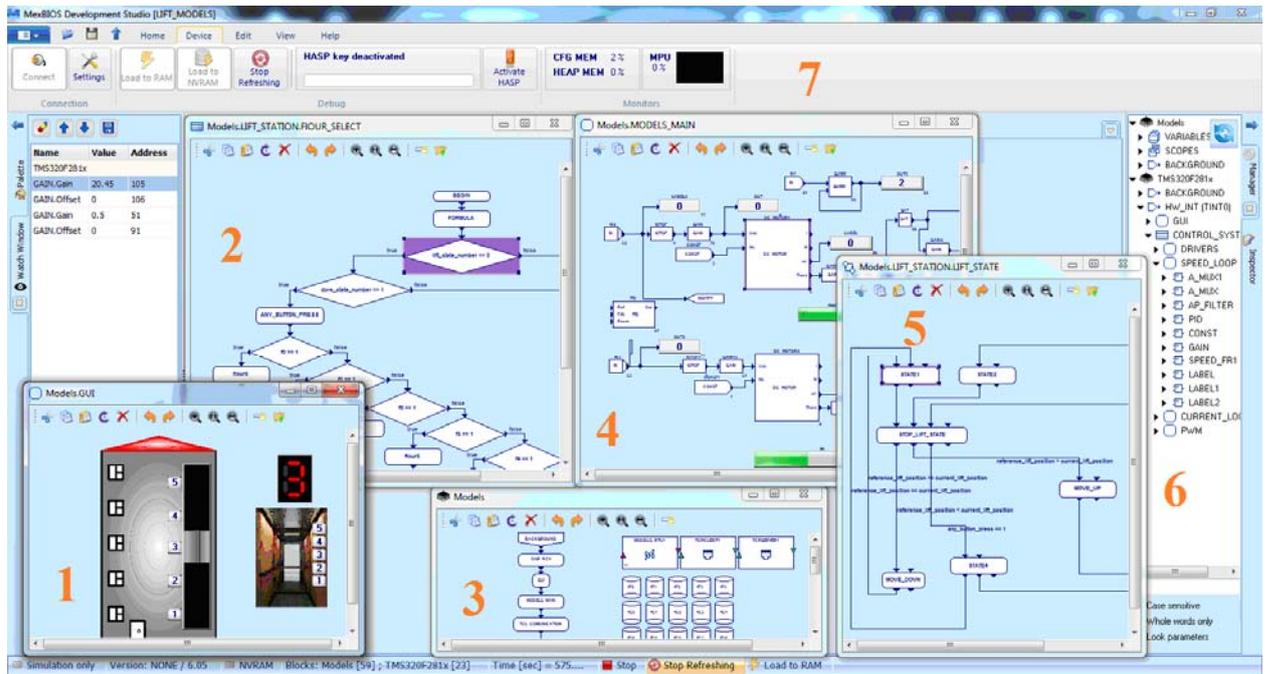


Рис. 1. Внешний вид САПР MexBIOS. Продемонстрирован экран работы по созданию системы управления лифтовой станцией: 1 – панель визуализации, 2 – поле алгоритмов, 3 – корневое поле для объявления переменных и событий, 4 – поле блок диаграмм, 5 – поле машины состояний, 6 – окно Менеджера проекта, 7 - меню системы.

## 2. Концепция и особенности системы MexBIOS

Концепция системы **MexBIOS** подразумевает наличие «стартового» проекта встроенного программного обеспечения для микроконтроллера (в роли которого может выступать и уже имеющиеся проекты пользователя системы). «Стартовый» проект, помимо собственных функций, выполняет запуск специализированного программного ядра MexBIOS, которое в свою очередь запускает элементы библиотеки в соответствии с правилами, определенными пользователем в процессе графического программирования на персональном компьютере. Фактически пользователь должен задать состав исполняемых элементов библиотеки, условия их запуска и потоки данных между входами-выходами примененных элементов библиотеки. Набор данных правил в дальнейшем будем называть «матрица». Фактически ядро системы есть интерпретатор, который вызывает заранее скомпилированные программные блоки в соответствии с созданной графическим способом программой. Интерпретация кода (пусть даже оперирующего с предварительно откомпилированными процедурами) естественно несколько замедляет исполнения программы, хранение «матрицы» в памяти данных уменьшает ее «полезную» емкость, но всё это может быть компенсировано через встроенный механизм генерации кода в текстовом виде после завершения процесса отладки.

Система MexBIOS имеет следующие особенности, позволяющие улучшить процесс разработки проектов:

1. Визуальное программирование позволяет выполнять разработку интеллектуальной составляющей системы управления инженерам, не имеющим профессионального опыта программирования. Проектирование на уровне блок-схем, алгоритмов, «машин состояний» является признанной тенденцией, упрощающей процесс разработки и документирования проекта.

2. Готовые шаблоны программного обеспечения, стандартных аппаратно-зависимых драйверов, модулей протоколов исключает необходимость освоения архитектуры примененного микроконтроллера.

3. Встраивание системы в существующие проекты позволяет сохранить предыдущие наработки и инвестиции в проекты. При этом добавление системы в проект позволяет резко увеличить адаптационные возможности устройства - фактически пользователь добавляет в устройство встроенную систему программирования.

4. Предварительное моделирование работы разрабатываемого кода совместно с моделями будущих объектов управления позволяет исключить глобальные ошибки проектирования и изучить работу программного кода исходя из потенциально возможных эксплуатационных ситуаций на ранних этапах проектирования.

5. Готовые программные процедуры библиотек системы и предоставляемых шаблонов для распространенных систем управления позволяет сократить инвестиции в НИОКР, ускорить этап лабораторных исследований.

6. Возможности создания графических «виртуальных» пультов управления позволяет упростить этап тестирования системы и ввода ее в эксплуатацию, ускорить процесс освоения устройства обслуживающим персоналом.

7. Принципиально новым решением системы MexBIOS является процесс графической разработки «на лету», когда в режиме отладки в реальном времени становится возможным не только менять значения переменных, но и алгоритмы исполнения кода.

8. Функция генерирования кода оставляет возможности для последующей оптимизации исходя из ресурсов процессора (при наличии такой потребности).

Рассмотрим основные этапы разработки системы на основе микроконтроллерного модуля TE-TMS320F28335 [1] с использованием САПР MexBIOS

### 3. Подключение MexBIOS к модулю TE-TMS320F28335

Одним из наиболее привлекательных процессоров реального времени является TMS320F2833x Texas Instruments. Компания Терраэлектроника - участник Texas Instruments Developer Network - предоставляет готовое решение на базе этого процессора - встраиваемый модуль TE-TMS320F28335 (рис.2). Он содержит всё необходимое для создания микроконтроллерного ядра системы управления электродвигателем в MexBIOS.

Описание модуля и документация доступны для скачивания на сайте разработчика модуля [1].

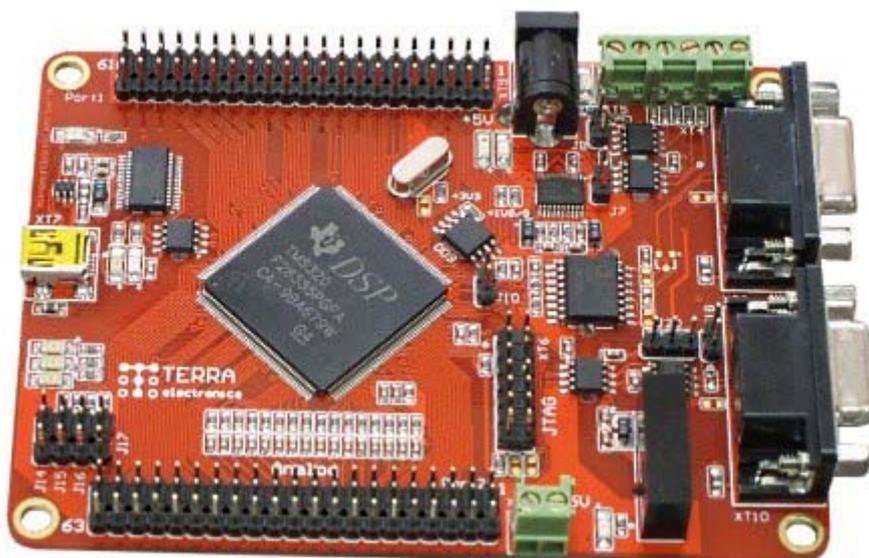


Рис. 2. Встраиваемый модуль TE-TMS320F28335

Для работы MexBIOS с конкретным контроллером необходимо провести минимальное количество действий, позволяющее вызвать ядро системы и осуществить коммуникации с компьютером, на котором установлена система MexBIOS:

1. Подключить библиотеку к имеющемуся проекту программного обеспечения устройства (далее - системному программному обеспечению). Размер библиотеки зависит от количества блоков, которое может быть откорректировано разработчиком, предоставляются возможности по добавлению в библиотеку собственных блоков.
2. Сконфигурировать интерфейс коммуникаций устройства для обеспечения связи с компьютером для обеспечения процесса загрузки программы пользователя и последующей отладки программы.
3. Настроить драйвер работы с EEPROM микросхемой для сохранения программы пользователя в устройстве.
4. Установить вызов интерпретатора по требуемому событию (например, прерыванию таймера).

После загрузки исполняемого файла в микроконтроллер пользователь связывается с ним посредством персонального компьютера, на котором создаются программы пользователя в графическом виде и после предварительного моделирования загружаются на исполнение в микропроцессор.

Стенд управления электродвигателем на основе модуля TE-TMS320F28335 приведен на рис.3

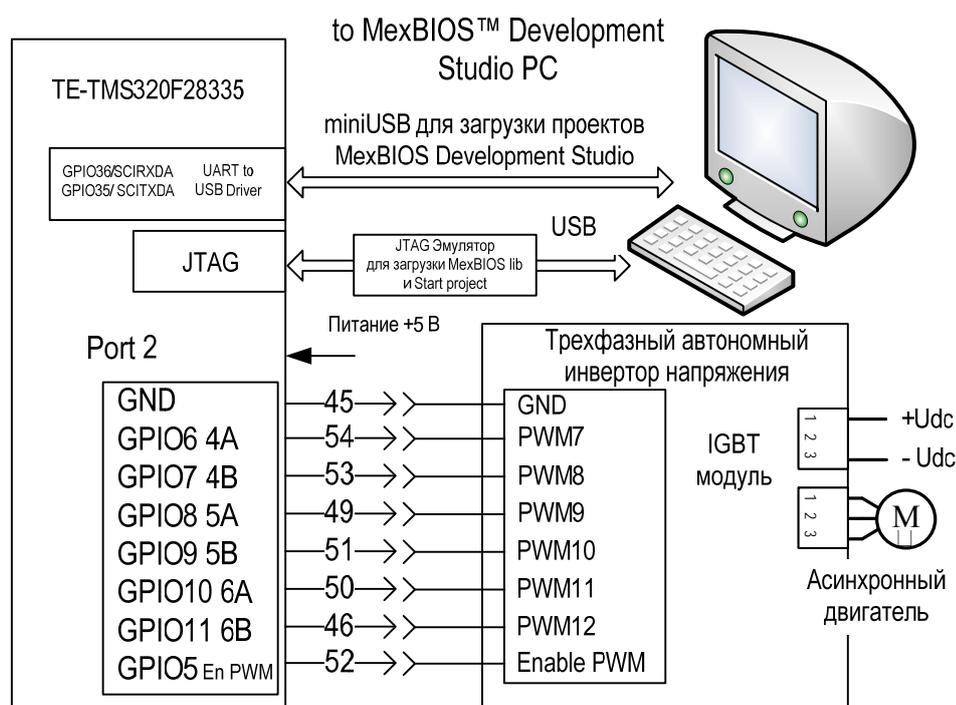


Рис. 3. Стенд управления электродвигателем на основе модуля TE-TMS320F28335 при программировании системой MexBIOS.

Для работы с MexBIOS™ Development Studio можно воспользоваться предоставляемым вместе со средой стартовым проектом. Загрузка стартового проекта и библиотеки производится через JTAG. Стартовый проект содержит настройки периферии и нужных для работы модулей. Ядро MexBIOS™ поддерживает обмен данными между стартовым проектом и проектом, созданным в системе MexBIOS™. Благодаря этому, целесообразно использовать проект пользователя с готовым ПО, а в системе MexBIOS™ реализовать какую-то часть программы, например, логику управления.

Чтобы подключить ядро MexBIOS в проект пользователя, необходимо выполнить следующие действия:

- 1) Подключить в модуле заголовочный файл, генерируемый системой MexBIOS™ Development Studio при компиляции библиотеки.

```
#include "MBS_Import.h" // MexBIOS library import header
```

- 2) В структуру конфигурации коммуникации добавить адрес, передающий данные для ядра MexBIOS.

```
// communication parameters
TDrvCommParams CommParams = {
    0, // UART identificator - SCIA
    1, // Slave address
    1152, // Baudrate / 100
    SYSCLK/8/115200-1, // Baudrate register value
    0, // Parity mode - None
    5, // Delay on byte reception (5/10000=0.0005 s)
    0, // Delay before transmission begins (0 s)
    HZ/1000, // Timing function frequency in kHz
    0, // Transmission control function
    (TDrvCommCallBack)MBS_GetDataAddr // Callback function
}
```

- 3) Вызвать в секции инициализации функцию «MBS\_Create» и передать параметры в ядро MexBIOS. После этого подать команду на запуск (\*MbsEnable=1).

```
// if MexBIOS enable execute function for
// setting it base parameters
KernelHandle = MBS_Create();

// set project version to display in the studio
*MbsAppVersion = VERSION;
```

- 4) В фоновой задаче вызвать функции задания конфигурации и исполнения фоновых задач

```
while(1){
    // if MexBIOS enable execute functions for
    // setting configuration and execute tasks
    MBS_Init(KernelHandle);
    MBS_IsrExecute(KernelHandle, 0);
    // execute function for communication
    SerialCommUpdate();
    // memory control function
    MemControl();}
```

- 5) В обработчике основного прерывания, например TINT0 - «CpuTimer0IsrHandler» вызвать обработчик периодических задач ядра

```
// execute function with interrupt source
MBS_IsrExecute(KernelHandle, 0x26);
MBS_MonExecute(KernelHandle);
```

- 6) По команде выполнить операцию записи конфигурации в энергонезависимую память

```

// send memory status to kernel
*MbsMemStatus = drvMem.Status;
// process memory command
if (!EraseStatus){
    if (*MbsMemCmd != 0) {MemErase(&drvMem); EraseStatus = 1;}
}
else
{if (*MbsMemCmd & 0x1) {MemWriteMtx(&drvMem); *MbsMemCmd &= ~0x1;}
else
if (*MbsMemCmd & 0x2) {MemWriteData(&drvMem); *MbsMemCmd &= ~0x2;}
if (*MbsMemCmd == 0) EraseStatus = 0;
}

```

- 7) Выполнить расчет загрузки системы и передать значение в ядро для отображения в среде разработки

```

// CPU load calculation
*MbsCpuLoad = (CPUL_GAIN * (-(Uint16)CpuTimer1Regs.TIM.all)) >> 21;

```

- 8) Выделить адресное пространство памяти программ и данных в командном файле для ядра MexBIOS, согласно настройкам Program memory в MexBIOS™ Development Studio.

После загрузки стартового проекта в память платы, можно подключиться с помощью системы MexBIOS™ через USB интерфейс. Покажем два простых примера реализации программы мигания светодиодом. Первый – использовать генератор прямоугольных сигналов, второй – создать событие срабатывания сигнала.

Реализация первого способа.

Вынести на поле набора блок EVENT. Настроить его как Hardware Interrupt. Выбрать прерывание TINT0 (объявленное в стартовом проекте), частота прерывания будет 10 кГц.

Подключить к блоку EVENT блок FORMULA. Внутри блока FORMULA собрать следующую схему:

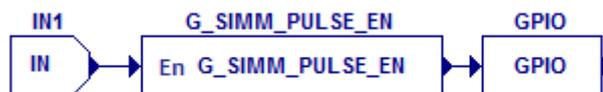


Рис. 4. Схема мигания светодиода с помощью встроенного генератора

Реализация второго способа.

Для этого необходимо добавить условие, по которому будет загораться/гаснуть светодиод. Такая схема представлена на рисунке 5.

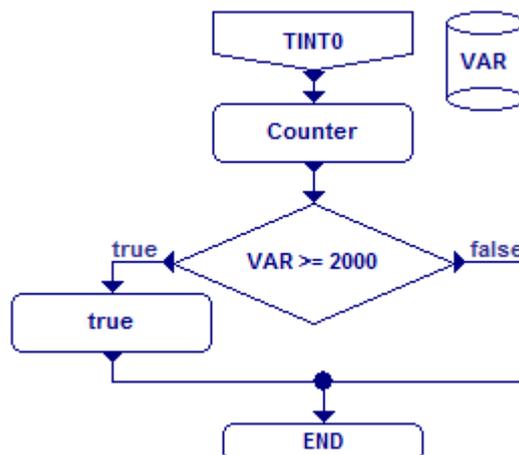


Рис. 5. Схема мигания светодиода с помощью события

Реализация условия начинается с добавления переменной VAR на поле набора (рис.6). Переменная VAR подключается к счётчику (счётчик состоит из двух блоков IN, блока

сумматора ADD и блока OUT), к которому также подключено условие зажигания светодиода (блок больше или равно GE и блок задания порога срабатывания CONST). Управление светодиодами происходит с помощью драйверов GPIO. Чтобы светодиоды мигали по очереди необходимо добавить блок инверсии сигнала LNOT.

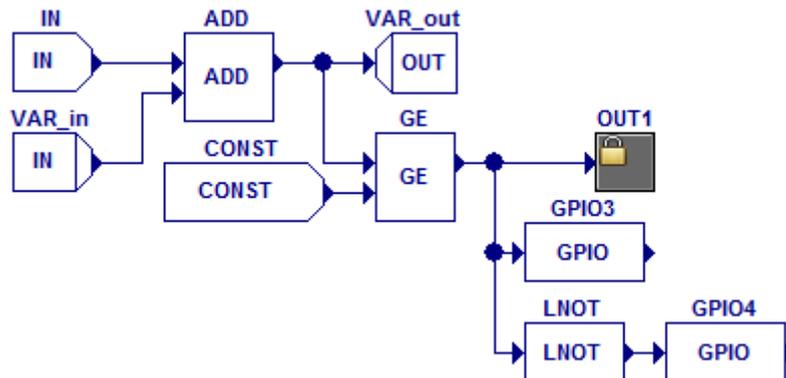


Рис. 6. Схема счётчика.

Для сброса счётчика необходимо добавить условие IF, при выполнении которого будет происходить сброс счётчика. В блок IF задать условие выполнения ветки true ( $VAR \geq 2000$ ).



Рис. 7. Схема сброса счётчика.

После загрузки файла конфигурации, на плате начнут мигать два светодиода GPIO3, GPIO4.

Благодаря применению метода графических библиотек становится возможным создавать программы для DSP без знания специфики программирования цифровых процессоров. Специалист, знающий структуру и настройки системы управления, может самостоятельно собрать графическую программу.

#### 4. Программирование модуля TE-TMS320F28335 в системе MexBIOS на стенде управления электродвигателем

Для создания простейшей скалярной системы управления на рис.8 необходимо проделать следующие действия:

1. Создать проект для TMS320F2833x.
2. Добавить блок EVENT (Палитра, группа блоков Embedded). Настроить блок следующим образом:
  - 2.1. Источник прерываний Source выбрать как Hardware interrupt (HW\_INT).
  - 2.2. Вектор прерываний выбрать TINT0 (либо выбрать вектор прерываний настроенный в стартовом проекте).
  - 2.3. Период задать равным периоду выбранного прерывания. Параметр Period необходим для моделирования, задаёт период выполнения схемы при моделировании.
  - 2.4. Параметр SimMode установить как Continuous. Настройка нужна для моделирования периодического прерывания. Если установлен пункт Enable, то присоединённые к EVENT блоки не будут выполняться. При установке One Short – присоединённые блоки выполняться один раз.
3. Добавить на поле набора формула FORMULA из Palette → Embedded.
  - 3.1. Подключить FORMULA к блоку EVENT. Переименовать блок FORMULA. Полученную структуру программы см. на рис.8.

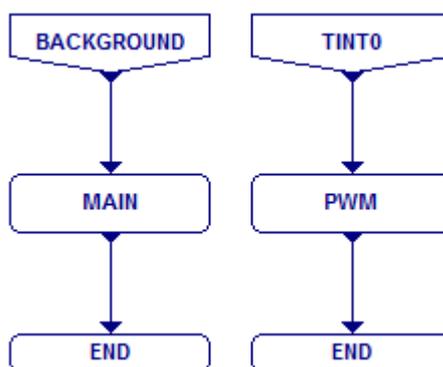


Рис.8. Структура программы

4. Перейти внутрь формулы PWM. Палитра изменит состав.
5. Добавить на поле набора блоки для скалярной системы управления:
  - 5.1. Блоки задания IN (Palette→Embedded).
  - 5.2. Из группы Palette→TMS320F2833x→Motor Control добавить блоки: кривая U/f V\_HZ, генератор пилообразного сигнала RAMPGEN, обратный преобразователь Парка IPARK, блок формирования вектора ШИМ SVGEN\_DQ.
  - 5.3. Из группы Palette→TMS320F2833x→Drivers добавить драйвер ШИМ PWM6, драйвер дискретной ножки процессора GPIO. При необходимости блоки можно копировать стандартными сочетаниями клавиш.
6. Собрать схему, аналогичную представленной на рис. 9. Чтобы вызвать Inspector параметров блоков, нужно выделить блок и нажать F11.
7. Параметры Ref\_V\_HZ: SampleTime=0.0001, BaseFreq=50, Gain=1, Offset=0. V\_HZ: LowFreq=0, HighFreq=1, MaxFreq=1, VoltMin=0, VoltMax=0.707.
8. Амплитуда вектора напряжения не может быть больше 1, поэтому ограничение двух составляющих обратного преобразователя Парка должно быть 0.707.
9. Блок SVGEN\_DQ не требует настройки.



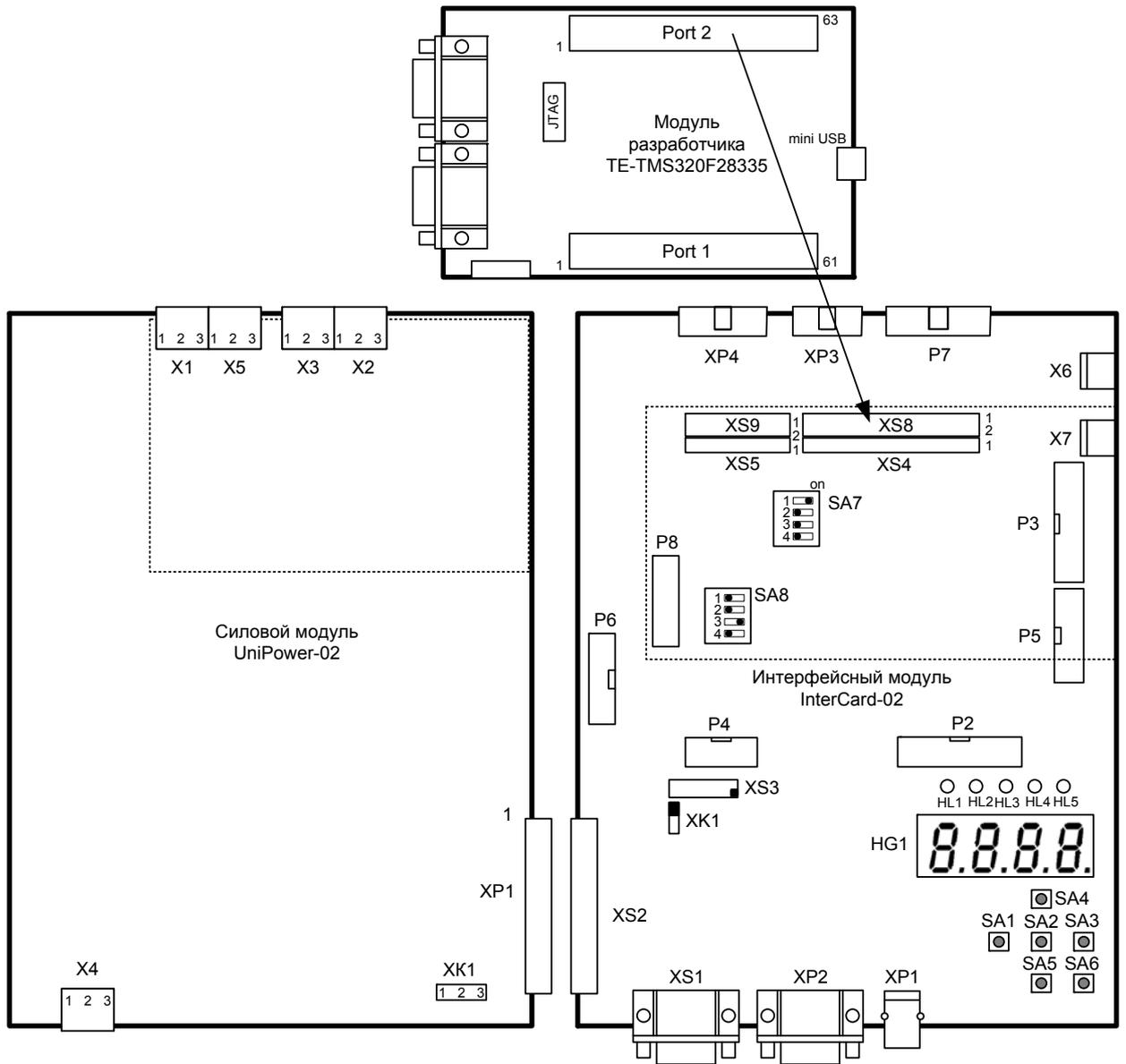


Рис. 10. Соединение TE-TMS320F28335 с MCB-02.

## Использование системы

Применение визуального программирования позволяет разрабатывать интеллектуальную начинку встроенной системы управления инженерам, не являющимися специалистами в области программирования. Проектирование на уровне блок-схем, алгоритмов, «машин состояний» является признанной тенденцией, упрощающей процесс разработки и документирования проекта.

Проектирование программного обеспечения в системе осуществляется с объявления переменных, используемых разрабатываемыми алгоритмами. Объявление переменных осуществляется путем выноса на рабочее поле системного блока VAR (см. рис.11). Переменные допускается выносить только на корневое поле проекта. В автоматически открывающемся окне инспектора необходимо задать имя переменной и тип (можно выбрать из списка как «плавающая запятая» либо «фиксированная запятая» заданного формата).

В дальнейшем, к переменной можно обращаться посредством блоков In и Out (соответственно считать значение и записать значение), предварительно указав в инспекторе свойств данного блока привязку к требуемой переменной.

Для прямого обращения в память существуют специальные блоки RD\_MEM (для считывания значения ячейки памяти) и WR\_MEM (для записи). При использовании данных блоков необходимо в окне инспектора свойств блока указать адрес в памяти данных, с которым будет работать данный блок.

Для удобства при обмене информации между основным (системным) программным обеспечением и интерпретатором можно подключить тар-файл, образованный при компиляции проекта встроенного программного обеспечения. Данный файл уже содержит информацию об адресах всех глобальных переменных, и список переменных будет отражен в инспекторе свойств в поле Address блоков RD\_MEM и WR\_MEM, через которые значения требуемых переменных становятся доступными для программы пользователя, что позволяет ему обращаться к переменным системного программного обеспечения.

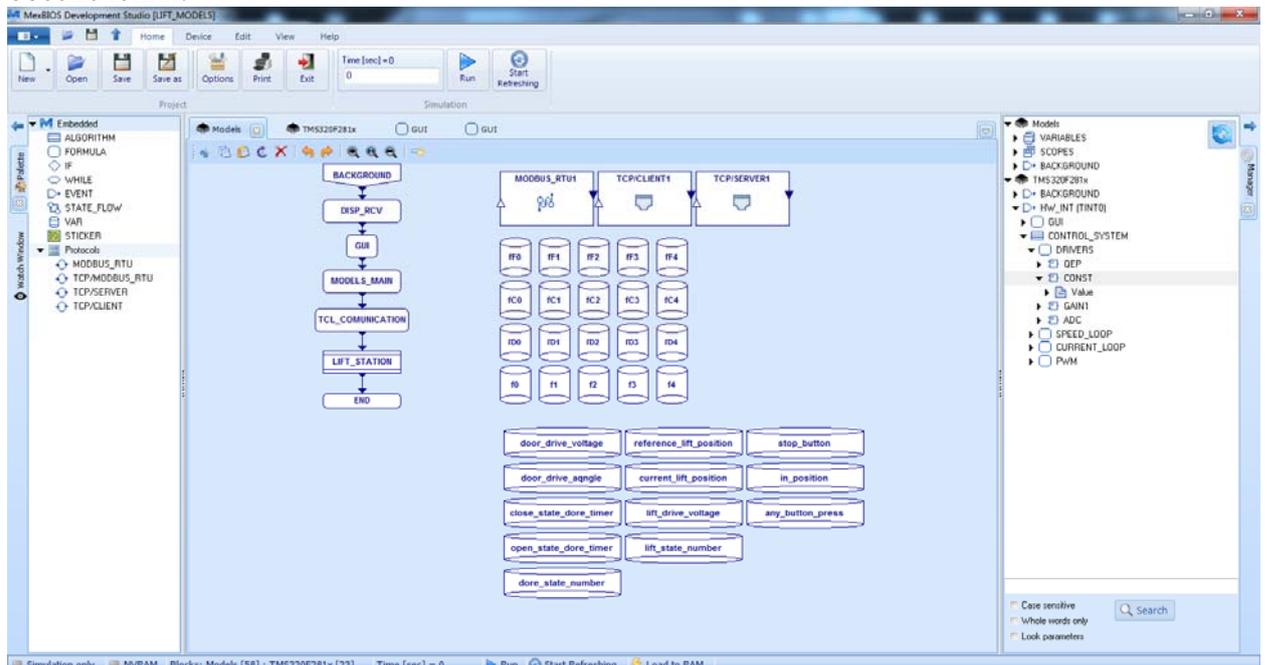


Рис. 11. Показано корневое поле для проекта системы управления лифтовой станцией. На экране находится алгоритм, вызывающий последовательно основные модули системы, объявлено использование протоколов коммуникаций (Modbus\_RTU для связи с системой управления электродвигателем главного движения, TCP/CLIENT1 и TCP/SERVER1 для

связи по сети Ethernet с компьютером диспетчера), в центре рабочего поле объявлены переменные.

Проектирование системы в среде осуществляется за счет прорисовки алгоритмов, в которых необходимо указать условия запуска формул (осуществляющие изменения значения переменных) либо следующих вложенных алгоритмов.

Формулы создаются путем прорисовывания блок-схем за счет использования готовых блоков, являющихся графическим отображением функции запрограммированной в микроконтроллер библиотеки. Пользователь должен установить требуемый набор блоков и соединить между собой их входы и выходы, обеспечивающие требуемые потоки данных, пример блок-схемы цифрового ПИД регулятора показан на рис.12.

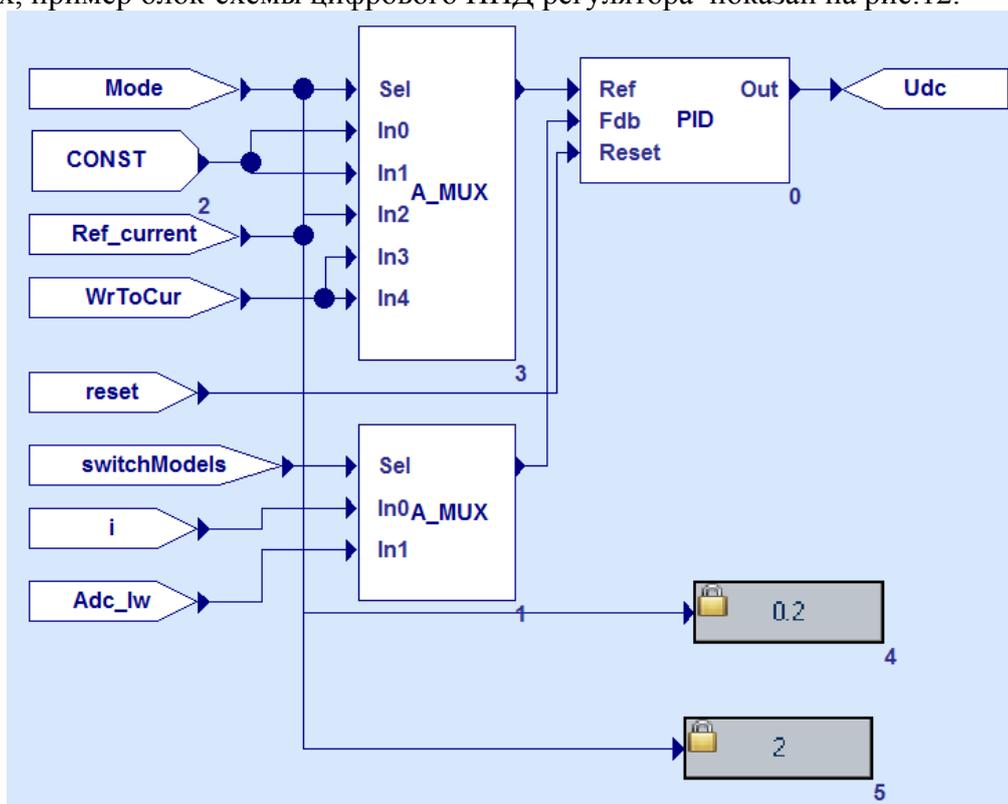


Рис. 12. Блок-схема цифрового ПИД регулятора. Блоки мультимплексора A\_MUX определяет источник получения сигнала обратной связи и задания на основе состояния внешних переменных

Каждый блок изначально создан с помощью текстового программирования, текст может быть просмотрен пользователем и при необходимости модифицирован (для этого потребуется перепрограммировать микроконтроллер в связи с изменением библиотеки, либо воспользоваться блоком текстового интерпретатора). На рис.13. показано окно текстового редактора конструктора блоков среды MexBIOS Development Studio, демонстрирующее создание блока нелинейной зависимости.

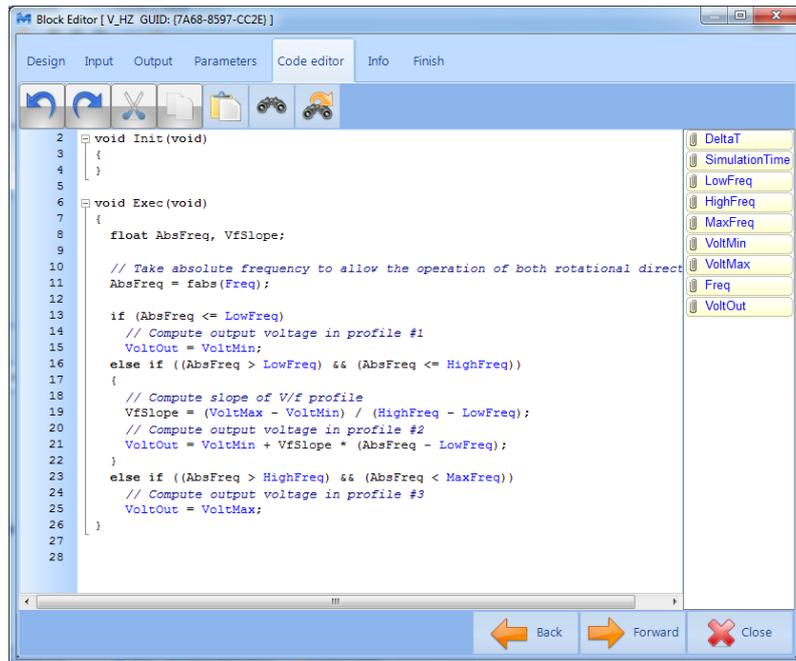


Рис. 13. Конструктор блоков системы, позволяющий задавать количество и тип входов/выходов и прописывать программный код в текстовом виде.

Алгоритмы могут быть вложены друг в друга согласно иерархии исполнения. Кроме условий запуска, можно организовать и циклическое исполнение выбранных участков (блок «While»). Фактически алгоритмы определяют условия запуска и порядок выполнения формул, в свою очередь выполняющих непосредственные действия над данными.

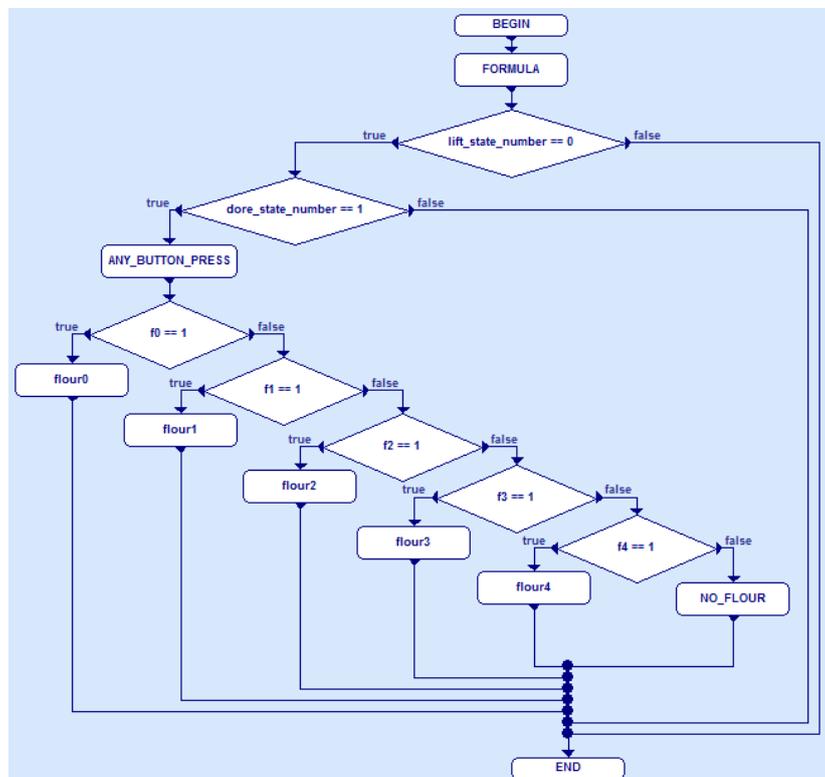


Рис. 14. Пример алгоритма в среде Mechatronics Development Studio. Элементы условного ветвления «if» (ромбы) на основании своих условий запускают либо другую ветку алгоритма, либо требуемую формулу (прямоугольник)



Поддержка протоколов коммуникаций – позволяет непосредственно из среды MexBIOS Development Studio связываться по стандартным протоколам коммуникаций типа Ethernet, Modbus и т.д. с устройствами, поддерживающими данные протоколы.

Подключение внешних приложений – позволяет среде обмениваться данными с другими приложениями.

Готовые шаблоны распространенных систем управления – позволяют быстро запустить управление электродвигателем требуемого типа.

## Заключение

Рассмотренная система эффективно позволяет программировать одновременно как в текстовом виде (задачи системного программного обеспечения и создания блоков), так и в различных парадигмах графического программирования. Встраивание системы в существующие проекты программного обеспечения позволяет значительно повысить функциональные свойства устройств, предоставить пользователю возможность настраивать не только параметры устройства управления, но и логику его работы. Наглядное графическое представление алгоритмов значительно упрощает их понимание и делает возможным проводить дальнейшие работы по оптимизации работы устройства, при изменении объекта управления или требований технологического процесса. Система прежде всего рассчитана для устройств, пользователи которых обладают минимально-достаточной квалификацией для создания собственных алгоритмов управления, которые разработчики устройства не могут заложить изначально вследствие отсутствия знаний об особенностях его применения для конкретных целей.

Для облегчения использования системы в неё добавлено большое количество документации и вспомогательных обучающих материалов. Система вызова справки разделена на 3 части

- 1) справки непосредственно по среде проектирования,
- 2) справка по библиотеке блоков,
- 3) справка по схеме (это pdf-файл, который прикладывается к созданному в среде проекту с таким же именем, как назван сам проект).

Дополнительно имеются видеоуроки, на которых показывается способы использования ключевых особенностей системы, а также целый ряд готовых примеров проектов, демонстрирующих её возможности, на русском и английском языках.

Систему допускается бесплатно применять для некоммерческих целей

Систему можно скачать с сайта производителя [2].

Система MexBIOS и библиотека для контроллеров TMS320F2833x идут в комплекте с модулем TE-TMS320F28335.

## Литература

1. Модуль TE-TMS320F28335 - <http://j.mp/TE-TMS320F28335>
2. [www.mechatronica-pro.com](http://www.mechatronica-pro.com)