

MexBIOS — интерпретатор графически-разработанных программ для микроконтроллеров

Александр КАРАКУЛОВ,
к. т. н.
Николай ГУСЕВ,
к. т. н.

В статье рассматривается расширение функциональных возможностей электронных устройств на базе микроконтроллеров за счет встраивания в них интерпретатора программ пользователя на основе системы MexBIOS. Программы пользователя могут быть реализованы в текстовом или графическом виде, например блок-схема, алгоритм, машина состояний. Интерпретатор подключается к имеющемуся программному обеспечению микроконтроллера в виде библиотеки и обеспечивает выполнение программы пользователя в реальном времени.

Введение

Применение систем программирования в свое время произвело настоящую революцию среди электронных устройств. За счет того, что стало возможным наращивать количество разнообразных функций посредством смены программного кода и без изменения аппаратной части, микропроцессорная техника быстро вошла во все сферы нашей повседневной жизни.

В настоящее время компании — флагманы электротехнической промышленности все чаще оснащают свои специализированные электронные устройства управления встроенными системами программирования, посредством которых пользователь может устанавливать собственные алгоритмы работы устройств промышленного применения — преобразователей частоты, сервоконтроллеров, модулей управления движением, блоков диагностики и сигнализации. Целесообразность интеграции функции программирования внутрь устройства очевидна и положительно оценена пользователями: не нужен дополнительный внешний контроллер, а значит, нет необходимости и в его программировании, отсутствуют проблемы совмещения габаритов и кабельных соединений, и пользователь в соответствии с конкретными требованиями может изменить логику работы своего устройства. Все это снижает стоимость запуска конечного оборудования, обслуживающего технологический процесс.

С помощью встроенной системы программирования, работающей, как правило, на языках программирования BASIC и Си, пользователь без привлечения разработчика может изменить алгоритм функционирования устройства для адаптации его к тре-

бованиям собственного технологического процесса. Осуществляется это посредством написанного пользователем программного кода, для которого в качестве входных данных принимается состояние дискретных и аналоговых входов, а также логических состояний и значений внутренних переменных, а в качестве выходных данных — задания на обработку технологической величины и значения коэффициентов и параметров настроек. При этом базовое (системное) программное обеспечение устройства остается неизменным, меняется только логический «интерфейс» взаимодействия с внешней средой.

В связи с тем, что встраивание систем программирования внутрь устройства практически не ведет к изменению аппаратной части и, более того, такие системы могут быть интегрированы в уже выпускающиеся устройства, для увеличения конкурентных преимуществ проектируемой и выпускаемой продукции определенную перспективу представляет создание универсальной системы исполнения программ пользователя на микроконтроллере.

Компания «НПФ Мехатроника-Про», участник Texas Instruments Developer Network, разработала систему MexBIOS, которая является интерпретатором программ пользователя и может быть интегрирована в проекты встроенного программного обеспечения различных микроконтроллерных устройств.

Описание системы программирования

Система MexBIOS предназначена для интеграции в 16- и 32-разрядные микроконтроллеры, она позволяет создавать программное обеспечение микроконтроллеров графическим и текстовым методом, в том числе

формировать события, формулы, алгоритмы, объявлять переменные и обращаться к ячейкам памяти. Система ориентирована на решение прикладных задач, в то время как обслуживание коммуникаций и периферийных устройств решается разработчиком на уровне системного программного обеспечения микроконтроллера. Система состоит из двух частей: аппаратно зависимой (ядро интерпретатора и библиотека функций) и среды программирования MexBIOS Development Studio.

Ядро интерпретатора и библиотека, содержащая скомпилированные функции, загружаются в память микроконтроллера и являются аппаратно зависимыми. Адаптация для конкретного типа микроконтроллера выполняется с помощью его стандартного компилятора. Вызов ядра интерпретатора осуществляется из системного ПО микроконтроллера, а вызов функций библиотеки осуществляет интерпретатор исходя из программы пользователя. В качестве функций библиотеки выступают операции арифметики и логики, вычисления цифровых регуляторов и фильтров, координатные и тригонометрические преобразования и т. д. (всего более ста стандартных функций). Следует отметить, что при этом обеспечивается высокая скорость исполнения интерпретатором программного кода за счет запуска уже предварительно скомпилированных функций.

Среда программирования MexBIOS Development Studio предназначена для создания графических программ пользователя с возможностью последующей загрузки в микроконтроллер. MexBIOS Development Studio запускается на персональном компьютере, эта среда позволяет предварительно моделировать работу проектируемого алгоритма совместно с моделями объектов управления

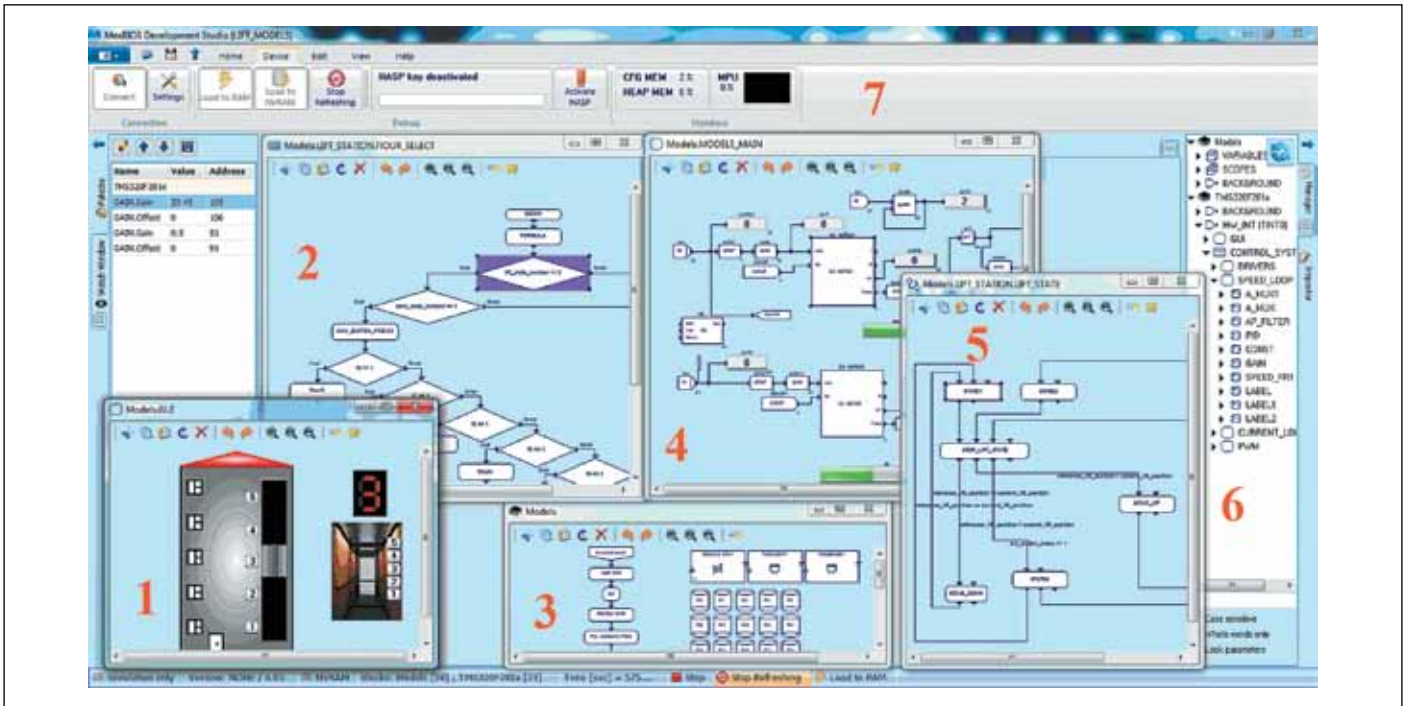


Рис. 1. Экран работы по созданию системы управления лифтовой станцией:

1 — панель визуализации; 2 — поле алгоритмов; 3 — корневое поле для объявления переменных, событий и используемых протоколов коммуникаций; 4 — поле блок-диаграмм; 5 — поле машины состояний; 6 — окно структуры проекта; 7 — панель меню

и проводить визуализацию управляемых процессов. При моделировании на персональном компьютере будет исполняться та же программа, которая затем будет загружена в микроконтроллер, при этом учитываются типы данных используемых переменных и ограничения на точность представления значений. В качестве моделей объектов, которыми должен управлять разрабатываемый код, могут выступать как имеющиеся в системе модели различных типов электродвигателей и механизмов, так и самостоятельно разработанные пользователем модели.

Внешний вид среды программирования показан на рис. 1, где продемонстрированы фрагменты проекта по созданию встроенного программного обеспечения для станции управления лифтами.

Среда программирования кодирует разработанные алгоритмы и блок-схемы в специальный файл, содержащий условия запуска требуемых функций библиотеки и направления передачи данных между функциями. Этот файл с программой загружается в микроконтроллер и выполняется интерпретатором.

Подключение системы к проекту

Для запуска системы на имеющемся микроконтроллере необходимо провести минимальное количество действий, позволяющее вызвать ядро системы и осуществлять коммуникации с компьютером, на котором установлена система MechBIOS. Для этого необходимо:

1. Подключить библиотеку к имеющемуся проекту программного обеспечения устройства (системному ПО). 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 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).

Более детально действия описаны в документации к системе, где также приведены примеры подключения системы для различных типов микроконтроллеров.

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

Использование системы

Применение визуального программирования позволяет инженерам, не являющимся специалистами в области программирова-

ния, разрабатывать интеллектуальную «начинку» встроенной системы управления. Проектирование на уровне блок-схем, алгоритмов, машин состояний является признанной тенденцией, упрощающей процесс разработки и документирования проекта.

Проектирование ПО в системе начинается с объявления переменных, используемых разрабатываемыми алгоритмами. Объявление переменных осуществляется путем выноса на рабочее поле системного блока VAR. На экране (рис. 2) находится алгоритм, вызывающий последовательно основные модули системы. Здесь показано, что используются протоколы коммуникаций (Modbus_RTU для связи с системой управления электродвигателем главного движения, TCP/CLIENT1 и TCP/SERVER1 для связи по Ethernet с компьютером диспетчера), в центре рабочего поле объявлены переменные. Переменные допускаются выносить только на корневое поле проекта. В автоматически открывающемся окне инспектора необходимо задать имя переменной и тип. (Можно выбрать из списка: «плавающая запятая» либо «фиксированная запятая» заданного формата.)

В дальнейшем к переменной можно обращаться посредством блоков In и Out (соответственно — считать 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 их использовании необходимо в окне инспектора свойств

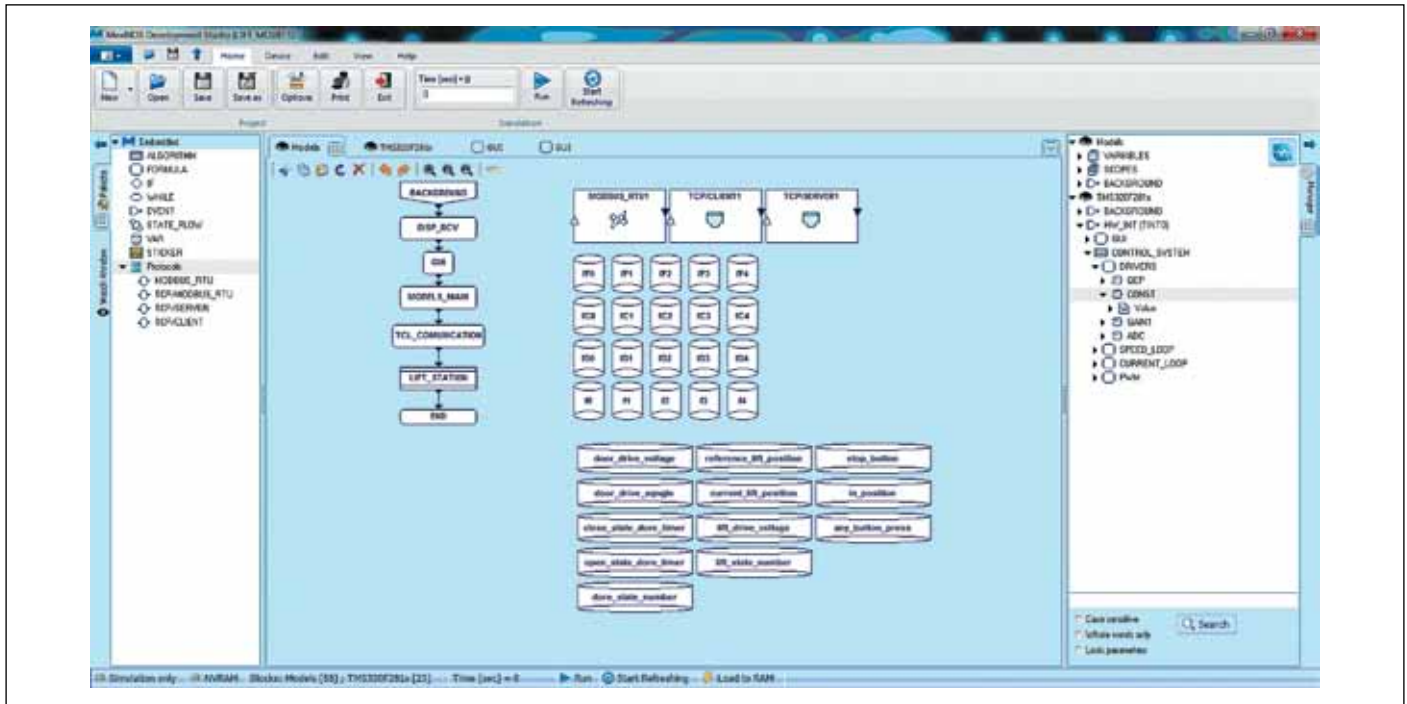


Рис. 2. Корневое поле для проекта системы управления лифтовой станцией

блока указать адрес в памяти данных, с которым будет работать этот блок.

Для удобства при обмене информации между основным (системным) ПО и интерпретатором можно подключить тар-файл, образованный при компиляции проекта встроенного программного обеспечения. Этот 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 доступны для программы пользователя, что

позволяет ему обращаться к переменным системного ПО.

Проектирование системы в среде осуществляется за счет прорисовки алгоритмов, в которых необходимо указать условия запуска формул, которые осуществляют изменение 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 требу-

емый набор блоков и соединить между собой их входы и выходы, обеспечивающие необходимые потоки данных. Пример блок-схемы цифрового ПИД-регулятора показан на рис. 3. Блок мультиплексора A_MUX определяет источник получения сигнала обратной связи и задания его на основе состояния внешних переменных.

Каждый блок изначально создан с помощью текстового программирования. Пользователь может просмотреть текст и при необходимости его модифицировать. (Для этого потребуются перепрограммировать микроконтроллер в связи с изменением

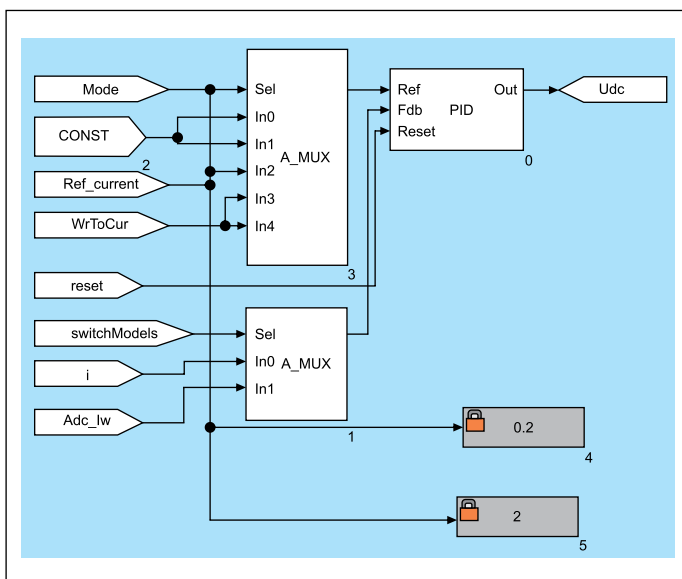


Рис. 3. Блок-схема цифрового ПИД-регулятора

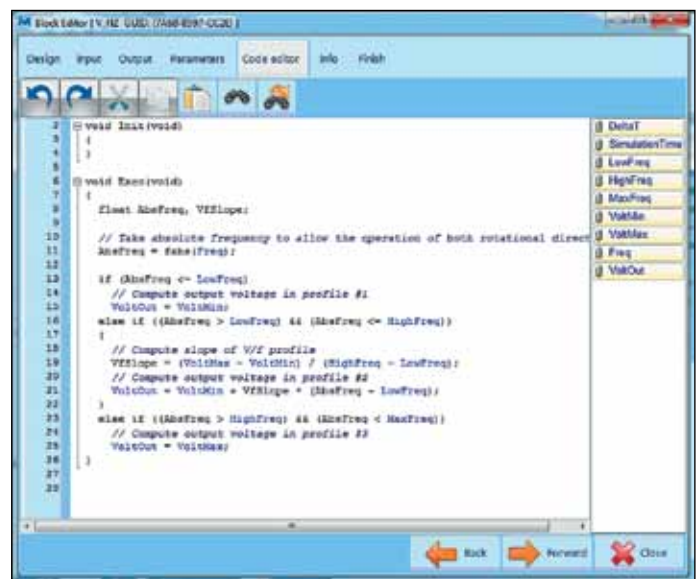


Рис. 4. Конструктор блоков системы, позволяющий задавать количество и тип входов/выходов и прописывать программный код в текстовом виде

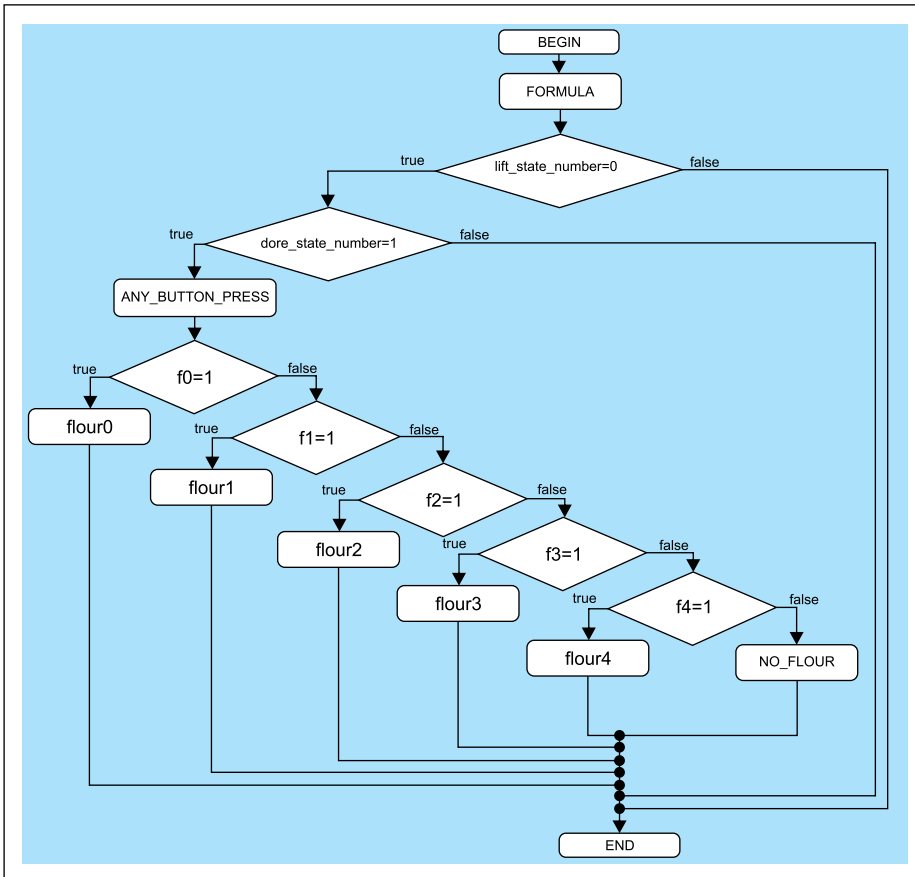


Рис. 5. Пример алгоритма в среде MexBIOS Development Studio

библиотеки либо воспользоваться блоком текстового интерпретатора.) На рис. 4 показано окно текстового редактора утилиты конструктора блоков среды MexBIOS Development Studio, демонстрирующее создание блока нелинейной зависимости.

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. Элементы условного ветвления if (ромбы) на основании своих условий запускают другую ветку алгоритма (рис. 5) или требуемую формулу (прямоугольник).

Машина состояний — это метод создания программ (автоматное программирование), который набирает все большую популярность. Пользователь должен определить основные логические состояния, в которых может находиться управляемая система (например, «Стоп», «Движение вверх», «Движение вниз» и т.д.), определить действия, характерные для каждого состояния (включить контактор, установить скорость и т.д.), и указать направления и условия перехода между состояниями. Этот метод позволяет легко наращивать функциональность алгоритма, он прост и нагляден. Система может создать

любое количество состояний (ограничение только по максимально свободной памяти микроконтроллера), с установлением индивидуальных условий перехода и детализацией действий для каждого состояния.

Для создания машины состояний необходимо вынести на рабочее поле нужное количество блоков состояний, соединить состояния между собой стрелками, ука-

зывающими направления перехода, разместить на стрелках условия перехода и в каждом состоянии установить соответствующие блок-схемы. На рис. 6 показана машина состояний для управления перемещением лифта.

Следует отметить, что MexBIOS Development Studio предоставляет множество дополнительных возможностей, позволяющих говорить о ней как о полномасштабной системе автоматизированного проектирования программного обеспечения. Большинство из них продемонстрированы в видеоуроках, прилагающихся к системе. Перечислим лишь некоторые из них:

- Генерация кода — позволяет на основе графически сформированной программы создать ее текстовое представление на языке Си, с последующим применением в системном ПО микроконтроллера.
- Вывод графиков — позволяет визуализировать изменение значения наблюдаемой переменной за заданный промежуток времени.
- Блоки драйверов — позволяют напрямую конфигурировать периферийные блоки микроконтроллеров (АЦП, ШИМ, дискретные пины процессора и т.д.).
- Работа с событиями — позволяет запускать алгоритмы исходя из аппаратных событий (например, прерывание таймера) и программных (например, при значении переменной выше заданного уровня).
- Создание интерфейсов управления — позволяет создавать графические (в том числе анимированные) интерфейсы управления для персональных компьютеров.
- Поддержка протоколов коммуникаций — позволяет непосредственно из среды MexBIOS Development Studio связываться по стандартным протоколам коммуникации типа Ethernet, Modbus и т.д. с устройствами, поддерживающими эти протоколы.

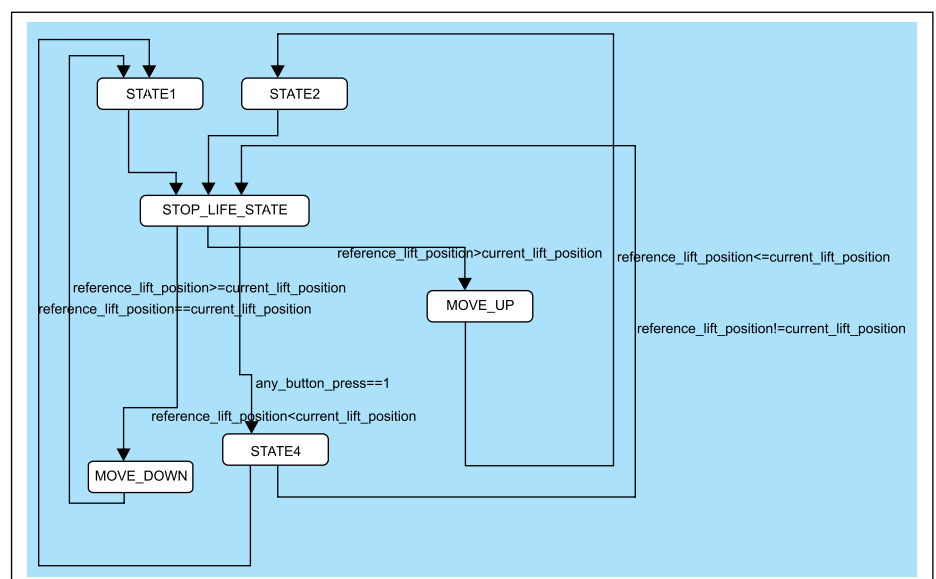


Рис. 6. Пример машины состояний, описывающей логику управления лифтом

- 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 дает возможность эффективно программировать как в текстовом виде (задачи системного програм-

многo обеспечения и создания блоков), так и в различных парадигмах графического программирования, причем одновременно. Встраивание системы в существующие проекты программного обеспечения позволяет значительно повысить функциональные свойства устройств. Пользователь получает возможность настраивать не только параметры устройства управления, но и логику его работы. Наглядное графическое представление алгоритмов значительно упрощает их понимание и позволяет проводить дальнейшие работы по оптимизации работы

устройства при изменении объекта управления или требований технологического процесса. Система прежде всего рассчитана для устройств, чьи пользователи обладают минимально достаточной квалификацией для создания собственных алгоритмов управления, которые разработчики устройства не могут заложить изначально из-за отсутствия знаний об особенностях его применения для конкретных целей.

Систему можно скачать с сайта производителя www.mexbios.com. ■