

Снижение энергопотребления, эксплуатационных расходов и времени вывода изделия на рынок благодаря более продвинутым методикам управления электродвигателями

Йон БЕРРОУЗ

Использование методик электронного регулирования может обеспечить значительную экономию электроэнергии и снижение стоимости разработки в большинстве приложений управления однофазными асинхронными электродвигателями.

Введение

Работающие двигатели потребляют примерно половину всей электроэнергии и используются во множестве бытовых приложений, таких как посудомоечные и стиральные машины, сушильные барабаны, нагреватели, системы вентиляции и воздушного кондиционирования. Все они в основной массе используют асинхронные однофазные двигатели переменного тока, которые работают в самом простом режиме: они либо полностью включены, либо полностью выключены.

Благодаря замене стандартного управления «включено-выключено» на управление скоростью вращения двигателя на этапе разработки таких приложений может быть значительно повышена эффективность использования энергии и снижены эксплуатационные расходы.

Большинство систем нагрева, вентиляции и кондиционирования воздуха (HVAC — Heating, Ventilation, and Air Conditioning) представляют собой яркий пример приложений, в которых все работает на полной скорости или выключено, без каких-либо промежуточных режимов. В этой статье мы продемонстрируем пример конструирования кондиционера воздуха с переменной скоростью. Решение этой задачи можно существенно упростить, если использовать типовую конструкцию (reference design) компании Microchip, которую она предлагает в качестве примера реализации технологии ProMPT (Programmable Motor Control Processor Technology, технология программи-

руемого процессора для управления двигателями) в «железе» и в основу которой положен оценочный набор Microchip для микроконтроллеров PIC18F2539.

Описание конструкции

Этот reference design был специально разработан для систем воздушного кондиционирования в системах HVAC, однако он представляет собой пример использования простого управления двигателем в любых приложениях. Микроконтроллер PIC18F2539 является flash-микроконтроллером с интегрированным ядром SPIMC (Single Phase Induction Motor Control — управление однофазными асинхронными электродвигателями). Ядро SPIMC позволяет осуществлять управ-

ление двигателем без обратной связи, формируя переменные напряжение и частоту в зависимости от кривой «напряжение — частота».

Структурная схема (рис. 1) представляет собой пример конструкции, обеспечивающей управление двигателем путем варьирования его скорости вращения. Reference Design для микроконтроллера PIC18F2539 может быть использован для управления двумя типами электродвигателей: с экранированными полюсами и с постоянными магнитами. Добавление других специфических возможностей возможно с помощью специальных плат расширения. На рис. 1 плата расширения к существующему приводу асинхронного двигателя переменного тока добавляет интерфейс пользователя и температурный датчик (выделены пунктирной линией).

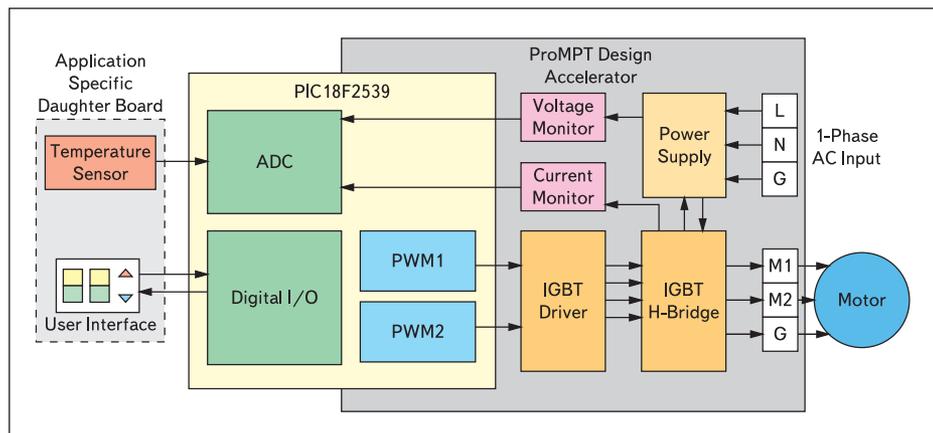


Рис. 1. Блок-схема компонентов приложения

Приведенная схема позволяет добавить управление с переменной скоростью вращения двигателя в кондиционер воздуха. Данная разработка не управляет нагревающими элементами — эта функция обеспечивается простым логическим управлением типа «включено–выключено» и реализуется с помощью крайне простых средств. Обычно термостат, установленный в помещении, задает целевую температуру. В этом приложении простой пользовательский интерфейс состоит из двух 8-сегментных светодиодных индикаторов и двух кнопок для задания целевой температуры.

Цепи управления двигателем построены на базе flash-микроконтроллера PIC18F2539 компании Microchip. Этот микроконтроллер имеет 8-разрядное RISC-ядро с 24-килобайтной памятью программ и 1,4-килобайтной памятью данных. Периферия включает в себя 6-канальный 10-разрядный АЦП, два модуля ШИМ, USART, I²C, SPI и до 20 цифровых линий ввода-вывода.

Для удобства разработки пользовательских приложений ядро управления двигателем встроено в микроконтроллер. Для обеспечения взаимосвязи ядра управления двигателем и пользовательского приложения используется набор функций API (Application Program Interface). Программа приложения может быть разработана, отлажена и выполнена при помощи среды разработки MPLAB IDE.

Определение температуры

В нашем устройстве температура воздуха измеряется при помощи датчика температуры TC1047A фирмы Microchip, который представляет собой преобразователь «температура — напряжение». Это исключает необходимость калибровки, которая обычно требуется в тех случаях, когда используются термопары. Датчик TC1047A представляет собой линейный датчик температуры, напряжение на выходе которого прямо пропорционально измеренной температуре. Он обеспечивает точное измерение температуры в диапазоне от -40 до +125 °C, что является более чем достаточным для большинства приложений отопления, вентиляции и кондиционирования воздуха.

Когда температура более чем на 5 °C ниже целевой температуры, кондиционер воздуха работает на полной мощности, а регулировка начинается, когда она находится в пределах 5 °C относительно целевой. Когда температура воздуха снижается, кондиционер увеличивает скорость и, таким образом, дает больший поток горячего воздуха в обогреваемое помещение.

Чтобы избежать частых переключений типа «включено — выключено», когда текущая температура находится вблизи целевой тем-

пературы, нижняя граница переменной скорости кривой отклика включает петлю гистерезиса. Для того чтобы кондиционер воздуха включался на минимальной частоте вращения двигателя, разница между текущей и целевой температурой должна превышать 1 °C. Однако для его выключения различие между указанными двумя температурами должно быть равно нулю. Поскольку многократное включение и выключение в такой системе исключено, она является менее шумной и позволяет существенно экономить электроэнергию и снизить создаваемые устройством помехи.

Ядро управления однофазным двигателем

Ядро управления двигателем на микроконтроллере PIC18F2539 (или любом другом микроконтроллере из этого семейства) использует метод переменной частоты для управления скоростью однофазного асинхронного электродвигателя. Два ШИМ-модуля микроконтроллера PIC18F2539 используются для синтеза тока синусоидальной формы, проходящего через обмотки двигателя. Ядро обеспечивает управление без обратной связи для диапазона частот от 15 до 127 Гц.

Скорость асинхронного электродвигателя является функцией частоты, смещения и числа полюсов в двигателе. Смещение предполагается постоянным в течение полезного рабочего цикла двигателя. Поскольку номинальная скорость основывается на числе полюсов и является фиксированной для конкретной конструкции двигателя, изменение частоты приложенного напряжения является единственным способом варьирования скорости вращения двигателя. Однако когда частота сигнала управления двигателем снижается, его импеданс также снижается, давая в результате более высокий ток тяги.

Приложенное к двигателю напряжение пропорционально обоим параметрам: частоте и току. Чтобы сохранить ток постоянным (на уровне тока полной нагрузки или ниже его), среднеквадратическое действующее значение напряжения на двигателе должно быть снижено в соответствии с частотой. Варьированием приложенного напряжения и частоты в постоянном их отношении друг к другу, скорость двигателя может варьироваться постоянным током. Поддержание этого постоянного отношения и является основной функцией ядра управления двигателем.

Микропрограммное обеспечение приложения

Ниже представлен список функций микропрограммного обеспечения для воздушного кондиционера с переменной скоростью вращения двигателя (задачи, поме-

ченные звездочкой, относятся к управлению асинхронным двигателем переменного тока):

1. Инициализация модуля управления двигателем*.
2. Установка соответствующей частоты напряжения для двигателя*.
3. Выполнение «вечного» цикла, который осуществляет выполнение следующих задач:
 - чтение значения температуры с датчика, подключенного к АЦП (канал RA0);
 - определение тока двигателя, напряжение шины DC и температуры нагревательного элемента;
 - чтение состояния кнопок и обеспечение защиты от дребезга контактов;
 - если нажата кнопка, то увеличить или уменьшить температуру, сохранить данные;
 - проверка на ошибки;
 - сравнить текущую температуру с целевой;
 - установить соответствующую частоту вращения двигателя*;
 - обновить показания светодиодного дисплея таким образом, чтобы он отображал текущую, актуальную на данный момент целевую температуру;
 - непрерывно выдавать управляющий сигнал для асинхронного двигателя переменного тока*.

Использование микроконтроллера PIC18F2539 и ядра на базе ProMPT существенно упрощает решение сложных задач. Функции API для ядра ProMPT делают разработку приложений управления двигателями и написание соответствующего кода более простыми. Постоянные, такие как отношение напряжения двигателя к частоте и коэффициент ускорения, определяются с помощью соответствующих функций интерфейса API и зависят от конкретного двигателя, используемого в приложении. Подробная блок-схема микропрограммного обеспечения приложения показана на рис. 2.

Программный интерфейс

Таблица синусов, хранящаяся в ядре ProMPT, используется как основа для синтеза постоянного тока с помощью модуля ШИМ. Значения таблицы читаются последовательно и масштабируются с учетом целевой рабочей частоты (скорости вращения) двигателя. Ввод частоты может быть осуществлен с одного из каналов АЦП, кроме того, в микроконтроллер может передаваться цифровое значение. Доступ к параметрам в модуле ProMPT осуществляется с помощью определенных методов API.

Память ядра содержит линейное отношение напряжения к частоте. Пользователи могут модифицировать коэффициенты в программе конкретного устройства для выполнения требований их приложения. Для этого

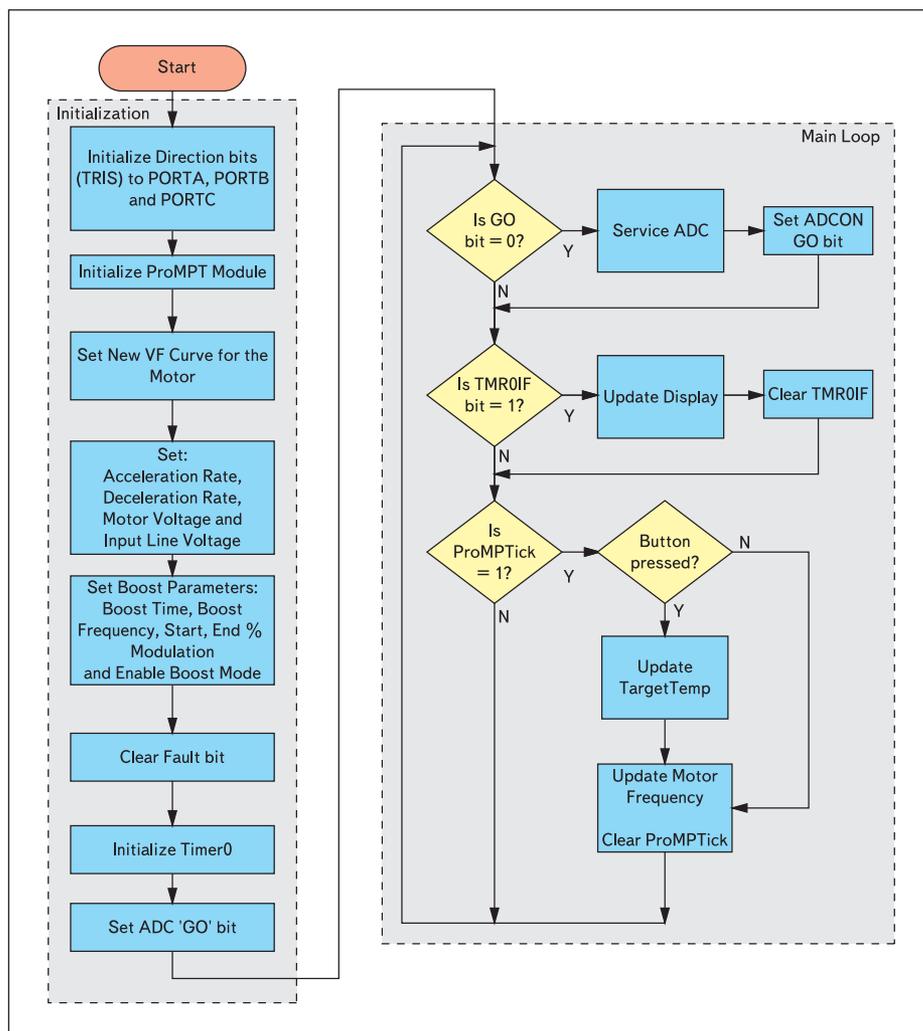


Рис. 2. Программа управления двигателем с использованием API ProMPT

используются API-метод *SetVFCurve(x,y)*, где x — частота, а y — степень модуляции напряжения шины DC. Обычно входная частота соответствует той точке на кривой отношения напряжения к частоте, которая обеспечивает 100%-ную модуляцию, согласованную с паспортным значением напряжения двигателя.

Использование ядра управления двигателем для разработки приложений

Ядро управления двигателем позволяет разрабатывать приложения пользователям, не имеющим знаний в теории функционирования двигателей и способов управления ими. Ключевые параметры ядра управления двигателем могут быть заданы и прочитаны при помощи соответствующих методов API. Приложение может иметь стек протоколов, показанный на рис. 3. Здесь методы API находятся между пользовательским приложением и ядром ProMPT и используются для обмена значениями параметров.

Приведенная на рис. 4 диаграмма изображает типовую программу управления двигателем. Согласно этому алгоритму двигатель будет запущен на частоте 20 Гц в течение 10 с, после чего частота увеличивается до 60 Гц со скоростью 10 Гц/с и остается на уровне 60 Гц в течение 20 с, и, наконец, двигатель останавливается. Методы API, комбинированные с параметрами двигателя и требованиями приложения, могут быть использованы для запуска двигателя на оптимальном крутящем моменте или КПД, а также для удовлетворения любого другого критерия.

Выводы

Управление скоростью может быть легко добавлено в систему отопления, вентиляции и кондиционирования воздуха при помощи оценочного набора компании Microchip, выполненного на базе микроконтроллера PIC18F2539. Его использование существенно упрощает конструкцию и снижает время вывода изделия на рынок для большинства приложений управления электродвигателями. Встроенное ядро управления двигателем позволяет пользова-

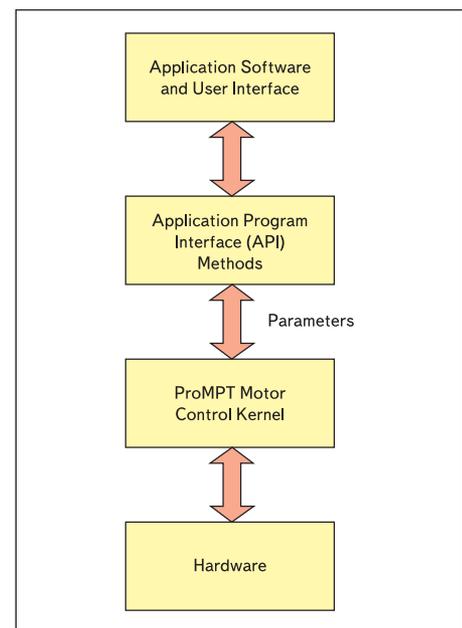


Рис. 3. Уровни стека архитектуры управления двигателем

```

Void main ()
{
  unsigned char i;
  unsigned char j;
  ProMPT_Init(0);           // Initialize the ProMPT block
  i = ProMPT_SetFrequency(10); // Set motor frequency // to 10Hz

  for (i=0; i 161; i++)      // Set cjunter for 10 sec @ // 1/16 sec per tick
  {
    j = ProMPT_Tick(void);   // Tick of 1/16 sec
    ProMPT_ClearTick(void);  // Clearing the Tick flag
  }

  ProMPT_SetAccelRate (10);  // Set acceleration rate // to 10 Hz/sec

  i = ProMPT_SetFrequency (60); // Set motor frequency // to 60 Hz

  for (i=0; i 161; i++);     // Set cjunter for 20 sev @ // 1/16 sec per tick // (2 loops of 10 sec each)
  {
    j = ProMPT_Tick(void);   // Tick of 1/16 sec
    ProMPT_ClearTick(void);  // Clearing the Tick flag
    j = ProMPT_Tick(void);   // Tick of 1/16 sec
    ProMPT_ClearTick(void);  // Clearing the Tick flag
  }
  i = ProMPT_SetFrequency(0); // Set motor frequency // to 0 Hz (stop)

  while(1);                 // End of the task
}

```

Рис. 4. Пример программы управления двигателем, использующей методы API ProMPT

телю с небольшими знаниями в области управления двигателями или при полном их отсутствии добавить возможность управления скоростью в многополюсные двигатели или двигатели с постоянными магнитами, что значительно расширяет возможности конечного продукта. ■

Литература

1. Microchip Technology AN861.
2. Microchip Technology PIC18F2539 Data Sheet.
3. ProMPT Developer Kit User's Guide.