

СРЕДСТВА МОНИТОРИНГА И ОТЛАДКИ СЛОЖНЫХ МНОГОПРОЦЕССОРНЫХ ПРИВОДОВ, ОБЪЕДИНЕННЫХ ПРОМЫШЛЕННОЙ СЕТЬЮ

При разработке собственной системы управления преобразователем частоты наибольшее время занимает создание программного обеспечения и его отладка. Имеется ввиду низкоуровневое программное обеспечение, управляющее непосредственно силовыми ключами инвертора, реализующим структуру управления двигателем, обрабатывающим сигналы аналоговых датчиков и т.п. Разрабатывается оно обычно на таких языках программирования, как C/C++ в средах разработки, предоставляемых производителем микроконтроллеров. Классический подход к отладке ПО, заключающийся в пошаговом проходе программы, чаще всего неприменим для микроконтроллерных систем управления, работающих в реальном времени. Программу, управляющую силовым инвертором, нельзя останавливать даже на короткий промежуток времени – это приведет в лучшем случае к аварийному останову системы, в худшем к выходу из строя силовых ключей инвертора. При работе двигателя можно только просматривать и изменять значения переменных, строить графики изменения их от времени (осциллограммы). Некоторые производители предоставляют средства для отладки микроконтроллеров в реальном времени без остановки управляющей программы микроконтроллера. Например, фирма Texas Instruments позволяет, в том числе для серии микроконтроллеров motor control, просматривать и редактировать любые переменные программы через собственную среду разработки и интерфейс JTAG, а также строить графики изменения величин от времени в режиме реального времени без остановки управляющей программы микроконтроллера. Это действительно удобно для однопроцессорного устройства, в котором доступен разъем JTAG во время работы электропривода. Однако часто силовые преобразователи закрыты и доступа к разъему интерфейса JTAG нет: вывести разъем JTAG на лицевую панель преобразователя невозможно из-за требований помехозащищенности, а из-за требований к конструктивному исполнению преобразователя быстрый доступ к его контроллеру внутри корпуса тоже невозможен. Для микроконтроллеров высокого класса с большим размером памяти и развитыми интерфейсами связи (Ethernet с TCP/IP стеком), подключенным ЖК дисплеем проблема отладки без доступа к JTAG стоит не так остро. Эти факторы заставляют разработчиков ПО специализированных микроконтроллеров, управляющих силовым электрооборудованием, искать новые пути отладки ПО без участия интерфейса JTAG. Средства для отладки сложных многопроцессорных систем привода должны позволять:

1. Просматривать и редактировать параметры устройства (коэффициенты регуляторов, уставки ограничений, задания и т.п.).
2. Просматривать в реальном времени текущие значения каких-либо величин (ток, напряжение, промежуточные величины системы управления, выходы регуляторов и т.п.)
3. Осциллографировать любую наблюдаемую величину или одновременно несколько величин – строить графики изменения от времени.
4. Перепрограммирование выбранных микроконтроллеров новой версией ПО.

Все эти задачи можно решить только при использовании какого-либо интерфейса связи микроконтроллера. Наиболее рационально для этого выбрать тот же интерфейс связи, которым объединяются микроконтроллеры всех устройств для решения выбранной технологической задачи. Из современных интерфейсов связи, доступных в специализированных микроконтроллерах, можно выделить CAN интерфейс, который широко применяется как в сфере сервоприводов, так и в автомобильной промышленности [1]. CAN позволяет организовать неразрывный доступ к шине нескольким устройствам одновременно, и обмениваться дополнительной сервисной информацией в фоновом режиме не мешая передаче основных пакетов данных. Существует два основных и наиболее распространенных протокола связи на его основе: CANopen и J1939. Оба поддерживают в своей архитектуре понятие номеров параметров (словаря объектов) с возможностью адресации индивидуально каждого параметра. Выбор в качестве базового CAN интерфейса и реализация одного из этих протоколов связи автоматически решает первые две задачи – просмотр и редактирование любых параметров микроконтроллерного устройства. Конечно, для этого потребуется реализовать драйвер протокола обмена как со стороны микроконтроллера, так и со стороны персонального компьютера (с использованием стороннего или собственного CAN интерфейса, например, CAN-USB), а также реализовать визуальную оболочку для удобной работы с драйвером. Драйвер CANopen [2, 3] для многих микроконтроллеров существует и в виде open-source решений, и в виде коммерческих библиотек. Для персонального компьютера существуют бесплатные утилиты для работы со словарем объектов CANopen, осуществляющие доступ к любому параметру словаря объектов CANopen на основе его SDO-сервиса. Однако их интерфейс чаще всего неудобен, поэтому рекомендуется разработка собственной программной оболочки для работы с CANopen драйвером. При этом кроме функции доступа к параметрам устройства в нее можно встроить функцию цифрового осциллографа, использовав для этой цели один из многочисленных компонентов для отрисовки графиков. Добавляя в такой программный осциллограф любые объекты

словаря CANopen и обновляя их в режиме реального времени через SDO-сервис, можно получить хорошую помощь в понимании процессов, протекающих при работе устройства. Однако частота обновления данных таким способом невелика и в зависимости от скорости передачи данных CAN сети может меняться от 50 до 500 Гц для одного параметра. Этого явно недостаточно, чтобы получать, например, осциллограммы мгновенных величин тока фаз электродвигателя или оценивать высокочастотные пульсации напряжения в звене постоянного тока преобразователя. Однако именно эти осциллограммы чаще всего и необходимы. Для получения таких осциллограмм используется метод предварительной записи требуемой осциллограммы внутри микроконтроллера с последующим её «скачиванием» на персональный компьютер пользователя. Загрузку осциллограммы можно выполнить итерационно по одной точке через SDO-сервис CANopen.

Когда все микроконтроллеры объединены общей сетью и обмениваются по ней некоторыми данными, удобно со стороны персонального компьютера осуществлять перехват таких данных, протоколировать в файл, а затем строить на их основе осциллограммы для последующего анализа. В протоколе CANopen для обмена неподтверждаемыми сообщениями с определенным интервалом служит сервис PDO, который также может быть обработан соответствующим драйвером на персональном компьютере, подключенном в общую сеть.

Следующим обязательным шагом является разработка собственного программатора (загрузчика) для обновления ПО микроконтроллера, работающего через CAN интерфейс [4]. Следует отметить, что многие производители предлагают для своих микроконтроллеров готовые загрузчики, позволяющие обновлять программу через различные интерфейсы связи – RS, CAN, SPI и т.п. Однако такие загрузчики обычно слишком примитивны и имеют собственный протокол передачи данных, не вписывающийся в выбранный протокол, использующийся в штатном режиме работы. Кроме того, очень желательно снабдить загрузчик такими функциями как разбивка всех передаваемых по сети данных на пакеты, верификация каждого пакета перед программированием в микроконтроллер, автоматическая перепосылка неудачно принятого пакета. Разработка такого программатора гораздо более сложная задача по сравнению с цифровым осциллографированием, но только с его использованием можно достичь приемлемых сроков отладки сложной распределенной микропроцессорной системы, где возможность обновления ПО через интерфейс JTAG ограничена.

Можно привести пример реального объекта, на котором приведенные выше средства отладки задействовались и были единственно-возможными. Производилась наладка электротрансмиссии транспортного средства, содержащей восемь преобразователей двигателей колес, два преобразователя генератора, контроллер верхнего уровня и DC/DC преобразователь аккумулятора. Преобразователи были выполнены во влагозащищенном корпусе и таким образом размещены в конструктиве транспортного средства, что не только исключало доступ к разъему JTAG, но и ограничивало доступ к преобразователю вообще. Реализация вышеупомянутых средств отладки позволила при движении машины протоколировать обмен сообщениями PDO между преобразователями и наблюдать осциллограммы изменения токов, частот вращения, напряжений приводов и т.п. Возможность снятия осциллограмм средствами микроконтроллера позволила отладить проблемы с устойчивостью систем управления в специфических режимах, когда частоты обновления данных PDO сообщениями не хватало для понимания сути проблемы. ПО преобразователей в конце работ содержало свыше 1000 объектов словаря CANopen, включающих в себя как параметры привода, переменные для наблюдения, так и служебные объекты. Программатор, работающий по сети CAN, позволял в течение дня по несколько раз обновлять ПО всех преобразователей и продолжать экспериментальную отладку уже с доработанными алгоритмами управления. Можно с уверенностью сказать, что только данные средства отладки позволили завершить проект и разобраться в сути всех протекающих в системе процессов. Никакая имитационная модель и лабораторный стенд не позволили бы подготовить программное обеспечение заранее, так как процессы и поведение оригинального объекта оказались уникальными. Таким образом, создание средств отладки, аналогичных рассмотренным, позволяют значительно ускорить разработку ПО, а в ряде случаев оказываются единственно-возможными для отладки.

Библиографический список

1. С.А.Третьяков, CAN на пороге нового столетия, Мир компьютерной автоматизации №2/99
2. CANopen Application Layer and Communication Profile, CiA Draft Standard 301, Version 4.02, Date: 13 February 2002
3. Анучин А.С., Сеть CAN и протокол верхнего уровня CANopen в современном электроприводе, Электричество, май 2008, с.23-29
4. Анучин А.С., Обновление программного обеспечения встроенных систем управления преобразователей частоты и источников питания по CAN-сети, Электрооборудование: эксплуатация и ремонт №3 // 2012, с.25-29.