Использование протоколов связи для удаленного управления электроприводом

Автор: В.Р. Губерна, А.В. Захаров, А.С. Евдокимов, Д.В. Бажутин
Источник: Перспективы развития электротехнических, электромеханических и энергосберегающих систем – 2017/ Материалы III международной научно-технической конференции студентов, аспирантов и молодых ученых. – Донецк, ДонНТУ – 2017, Том 2, с. 198-203.

Аннотация

В.Р. Губерна, А.В. Захаров, А.С. Евдокимов, Д.В. Бажутин Использование протоколов связи для удаленного управления электроприводом.

В работе рассматриваются вопросы удаленного управления электроприводом с использованием протоколов связи SPI и I2C. Приведен сравнительный анализ рассматриваемых протоколов связи, разработан алгоритм взаимодействия микроконтроллеров.

В настоящее время все более актуальным становится вопрос удаленного управления электроприводами в связи с невозможностью установки управляющего устройства в непосредственной близости отрабочего механизма. При этом силовую часть стараются расположить ближе к приводу, а управляющие сигналы на нее передавать удаленно. В системах малой автоматизации и механизмах, управляемых отнедорогих контроллеров, например, с архитектурой AVR или ARM,используются протоколы связи RS232, SPI и I2C.

I2C шина является одной из модификаций последовательных протоколов обмена данных [1]. В стандартном режиме обеспечивается передача последовательных 8-битных данных со скоростью до 100 кбит/с, и до 400 кбит/с в быстром режиме. Для осуществления процесса обмена информацией по I2C шине, используется всего два сигнала: линия данных SDA (линия синхронизации) и SCL (линия тактов). Простая двухпроводная последовательная шина I2C минимизирует количество соединения между устройствами.

Каждое устройство распознается по уникальному адресу и может работать как передатчик или приёмник, в зависимости от своего назначения.

Кроме того, устройства могут быть классифицированы как ведущие и ведомые при передаче данных. Ведущий – это устройство, которое инициирует передачу данных и вырабатывает сигналы синхронизации. При этом любое адресуемое устройство считается ведомым по отношению к ведущему.

Исходя из спецификации работы шины, в каждый отдельный момент в шине может быть только один ведущий, а именно то устройство, которое обеспечивает формирование сигнала SCL шины. Ведущий может выступать как в роли ведущего-передатчика, так и ведущего-приемника. Тем не менее – шина позволяет иметь несколько ведущих, накладывая определенные особенности их поведения в формировании сигналов управления и контроля состояния шины. Возможность подключения более одного ведущего к шине означает, что более чем один ведущий может попытаться начать пересылку в один и тот же момент времени.

Процедура обмена начинается с того, что ведущий формирует состояние СТАРТ – ведущий генерирует переход сигнала линии SDA из ВЫСОКОГО состояния в НИЗКОЕ при ВЫСОКОМ уровне на линии SCL.

Процедура обмена завершается тем, что ведущий формирует состояние СТОП – переход состояния линии SDA из низкого состояния в ВЫСОКОЕ при ВЫСОКОМ состоянии линии SCL.

Для подтверждения приема байта от ведущего-передатчика ведомым-приемником в спецификации протокола обмена по шине I2C вводится специальный бит подтверждения, выставляемый на шину SDA после приема 8 бита данных.

Таким образом, передача 8 бит данных от передатчика к приемнику завершаются дополнительным циклом (формированием 9-го тактового импульса линии SCL), при котором приемник выставляет низкий уровень сигнала на линии SDA, как признак успешного приема байта.

Подтверждение при передаче данных обязательно. Соответствующий импульс синхронизации генерируется ведущим. Передатчик отпускает (ВЫСОКОЕ) линию SDA на время синхроимпульса подтверждения. Приёмник должен удерживать линию SDA в течение ВЫСОКОГО состояния синхроимпульса подтверждения в стабильном НИЗКОМ состоянии.

Если в пересылке участвует ведущий-приёмник, то он должен сообщить об окончании передачи ведомому-передатчику путем неподтверждения последнего байта. Ведомый-передатчик должен освободить линию данных для того, чтобы позволить ведущему выдать сигнал СТОП или повторить сигнал СТАРТ.

Каждое устройство, подключённое к шине, может быть программно адресовано по уникальному адресу.

Процедура адресации на шине I2C заключается в том, что первый байт после сигнала СТАРТ определяет, какой ведомый адресуется ведущим для проведения цикла обмена. Исключение составляет адрес Общего вызова, который адресует все устройства на шине. Когда используется этот адрес, все устройства в теории должны послать сигнал

подтверждения. Однако, устройства, которые могут обрабатывать общий вызов, на практике встречаются редко.

Первые семь битов первого байта образуют адрес ведомого. Восьмой, младший бит, определяет направление пересылки данных. Ноль означает, что ведущий будет записывать информацию в выбранный ведомый. Единица означает, что ведущий будет считывать информацию из ведомого.

После того, как адрес послан, каждое устройство в системе сравнивает первые семь бит после сигнала СТАРТ со своим адресом. При совпадении устройство полагает себя выбранным как ведомый-приёмник или как ведомый-передатчик, в зависимости от бита направления.

В общем виде процесс обмена по шине от момента формирования состояния СТАРТ до состояния СТОП показан на рисунке 1.

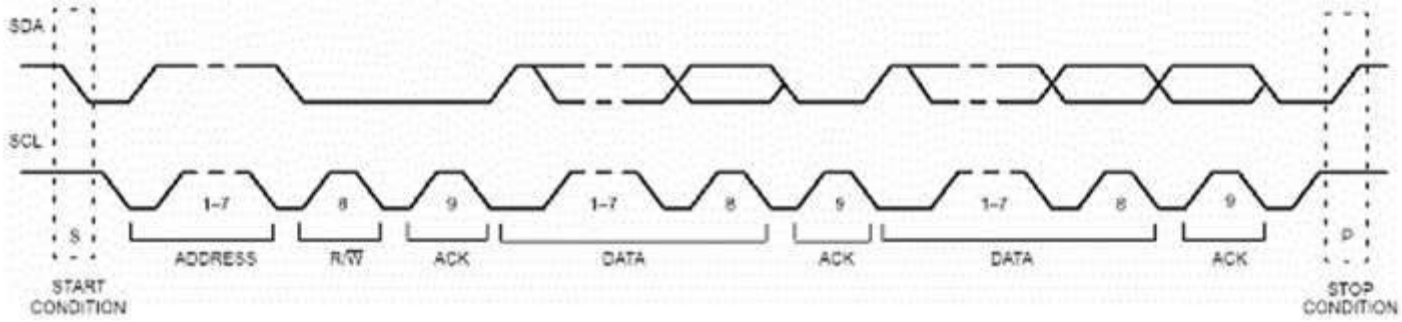

Рисунок 1 – Процесс обмена данными по шине I2C

Аббревиатура SPI означает Serial Peripheral Interface или в русском варианте последовательный периферийный интерфейс [2]. Название говорит само за себя, данный интерфейс используется для работы с различными периферийными устройствами. С технической точки зрения SPI – это синхронная четырёхпроводная шина. Она представляет собой соединение двух синхронных сдвиговых регистров, которые являются центральным элементом любого SPI устройства. Для соединения используется конфигурация ведущий/ведомый. Только ведущий может генерировать импульсы синхронизации. В схеме всегда только один ведущий (в отличие от той же шины I2C, где возможен вариант с более чем одним ведущим), количество ведомых может быть различно. В общем случае выход ведущего соединяется со входом ведомого, и наоборот, выход ведомого соединяется со входом ведущего. При подаче импульсов синхронизации на выход SCK, данные выталкиваются ведущим с выхода MOSI, и захватываются ведомым по входу MISO. Таким образом, если подать количество импульсов синхронизации, соответствующее разрядности сдвигового регистра, то данные в регистрах обменяются местами. Отсюда следует, что SPI всегда работает в полнодуплексном режиме. А вот нужны ли нам данные, полученные от устройства при записи какого-либо параметра, это уже другой вопрос. Часто бывает, что

данные, полученные от устройства при записи в него данных, являются мусором, в таком случае их просто игнорируют, но мы их получим вне зависимости от нашего желания.

Контроллер SPI, как правило, может работать как в режиме ведущего, так и в режиме ведомого.

Существует простейший способ включения SPI устройств. Такой способ показан на рисунке 2.

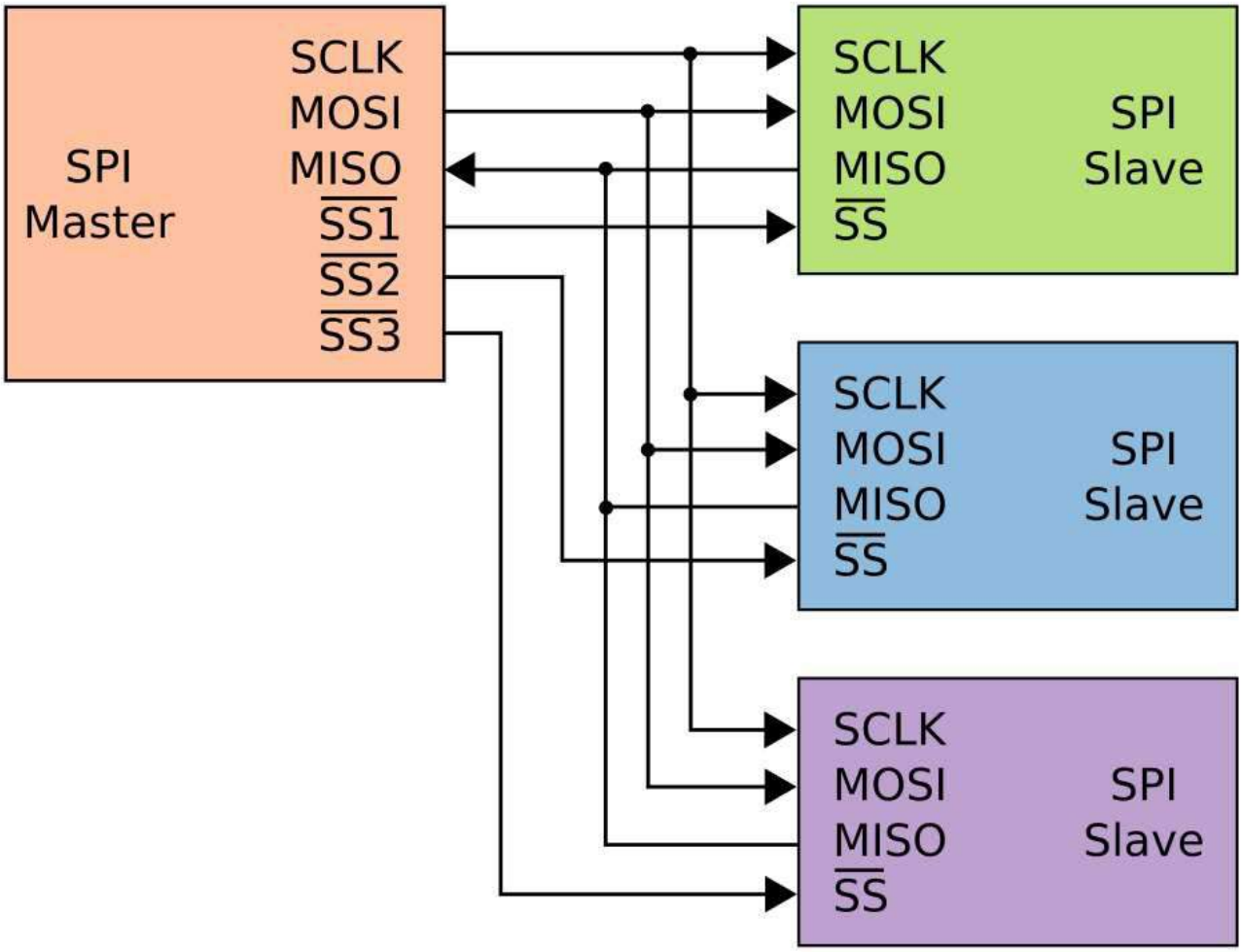

Рисунок 2 – Способ включения SPI с параллельным подключением ведомых устройств

В данном случае к ведущему все ведомые подключаются параллельно, за исключением сигнала выбора ведомого (~CS). Для каждого ведомого необходим отдельный сигнал выбора ведомого (на рисунке они обозначены как SSx). Для сигналов выбора ведомого могут использоваться как специально предназначенные для этого выходы SPI-контроллера, так и порты ввода/вывода общего назначения (GPIO) микроконтроллера.

Два проводника используются для передачи данных, один для подачи тактовых импульсов и по одному сигналу выбора ведомого для каждого из ведомых.

1. MOSI – Master Output, Slave Input (выход ведущего, вход ведомого). Данный сигнал предназначен для последовательной передачи данных от ведущего к ведомому. Также может называться SDO, DO и т.п.;
2. MISO – Master Input, Slave Output (вход ведущего, выход ведомого). Данный сигнал предназначен для последовательной передачи данных от ведомого к ведущему. Может называться SDI, DI и т.п.;
3. SCK – Serial Clock (сигнал синхронизации). Используется для синхронизации при передаче данных. Также может иметь название SCLK, CLK и др.;
4. ~CS – Chip Select (выбор микросхемы). С помощью данного сигнала происходит активация ведомого устройства. Обычно он является инверсным, то есть низкий уровень считается активным. Иногда его называют ~SS (Slave Select, рус. выбор ведомого).

Приведем пример использования протокола I2C.

Пусть одна плата Arduino Due будет системой управления (с кнопками для управления электроприводом), т.е. ведущим устройством. С этого ведущего устройства будут передаваться данные ведомому, которое является СПР скорости ДПТ. Таким образом, при помощи I2C-коммуникации можно настроить две платы Arduino Due таким образом, чтобы они делились информацией по принципу ведущая плата отсылает/ведомая плата считывает. Ведущая плата Arduino Due отсылает сигнал на пуск, торможение и реверс двигателя, а также входной сигнал ЗИ (заданную скорость), а ведомая принимает эти данные.

На рисунке 3 представлена схема подключения ведущей и ведомой платы Arduino Due. Контакты SDA и SCL на ведущей плате Arduino подключены к контактам SDA и SCL на ведомой Arduino. Кроме того, контакты с землей (GND) у обеих плат тоже подключены друг к другу. К ведущей плате подключены кнопки управления со световой индикацией, а к ведомой – двигатель.

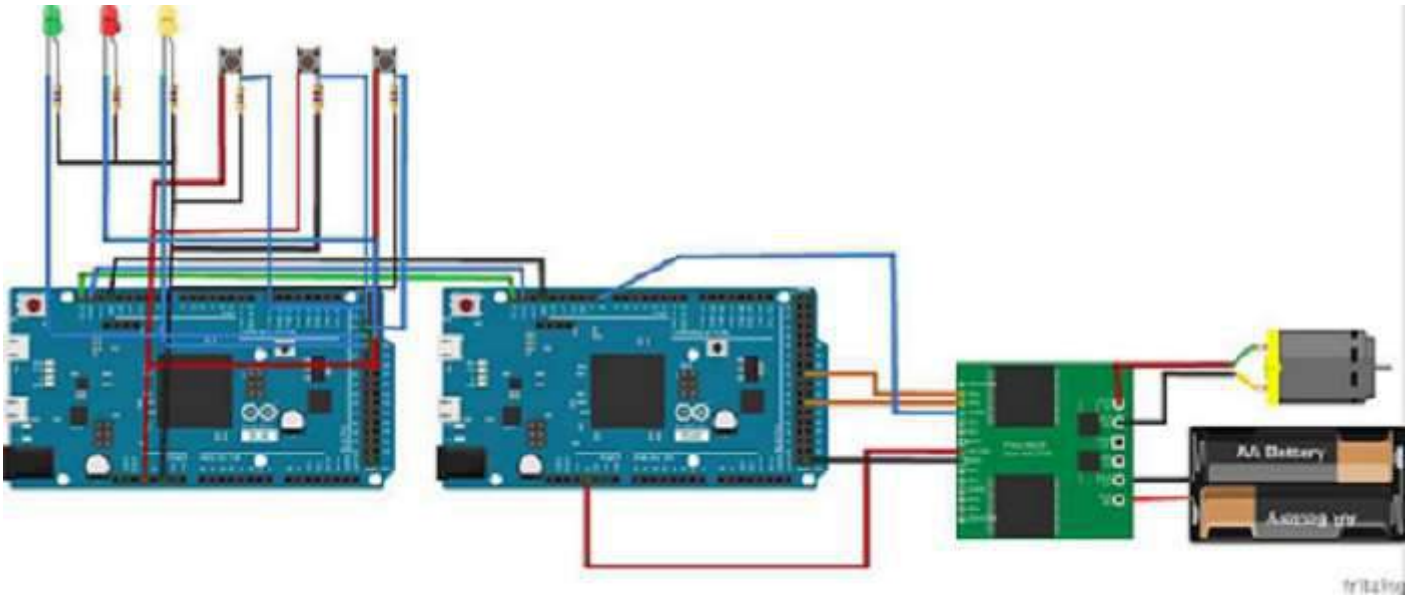

Рисунок 3 – Схема подключения ведущей и ведомой платы Arduino Due

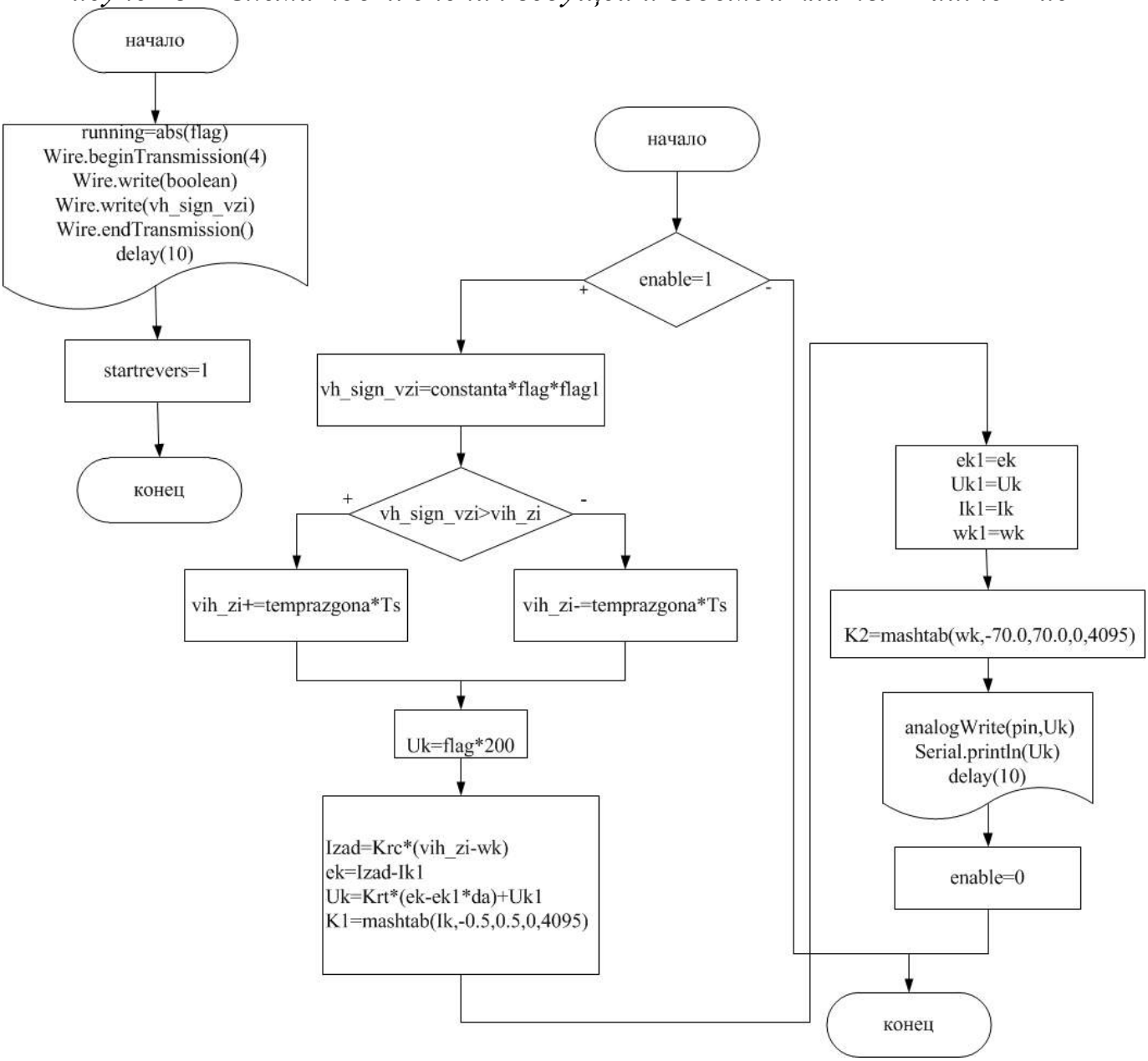

Рисунок 4 – Алгоритм программы ведущей и ведомой платы Arduino Due

Выводы

В отличие от протокола связи SPI, шина I2C остается двухпроводной, независимо от количества подключенной к ней микросхем и является более стандартизированной, а также к ней возможно подключить несколько ведущих микросхем.

Протокол SPI является более простым в плане программной реализации и на физическом уровне, что обуславливает надежность и быстродействие передачи данных. Все линии шины SPI являются однонаправленными.

Список использованной литературы

1. Описание шины I2C. http://www.itt-ltd.com/reference/ref_i2c.html
2. Обзор шины SPI и разработка драйвера ведомого SPI устройства для embedded Linux (Часть первая, обзорная) https://habrahabr.ru/post/123145/