

Обработка сигналов шестиканального аналого-цифрового преобразователя 1273ПВ19Т под управлением микроконтроллера 1986ВЕ1Т

Марков Игорь Николаевич, инженер-программист
Клюева Светлана Федоровна, кандидат технических наук, ведущий инженер-программист
АО «Аэроэлектромаш» (г. Москва)

Аннотация. В статье описан алгоритм управления и сбора данных с аналого-цифрового преобразователя 1273ПВ19Т посредством последовательной передачи с временным мультиплексным разделением каналов под управлением микроконтроллера 1986ВЕ1Т.

Ключевые слова: аналого-цифровой преобразователь, синхронная передача, каналы передачи, мультиплексирование каналов.

Введение

Шестиканальный 16-разрядный АЦП 1273ПВ19Т является функциональным аналогом AD73360 производства фирмы Analog Devices, США [1, 2]. Обмен данными между 1273ПВ19Т и процессором цифровой обработки сигналов осуществляется последовательной синхронной передачей данных, причём ведущим устройством является АЦП, а не процессор.

Микроконтроллер 1986ВЕ1Т построен на базе высокопроизводительного RISC ядра ARM Cortex-M и способен работать на частоте до 144 МГц. Периферия микроконтроллера включает в себя такие интерфейсы, как USB, Ethernet, UART, SPI, а также авиационные интерфейсы ГОСТ 18977-79 (ARINC) и ГОСТ Р520702003 [3].

Для программирования и работы микроконтроллера 1986ВЕ1Т используется «Комплект отладочный для микросхемы 1986ВЕ1Т», поставляемый компанией АО «ПКК Миландр». Программирование и работа с отладочным комплектом происходит в среде разработки CodeMaster-ARM(Phyton), для прошивки микроконтроллера используется USB JTAG адаптер JEM-ARM-V2(Phyton) [4].

В рамках этой статьи рассмотрены следующие вопросы:

- устройство и обмен данными с АЦП 1273ПВ19Т;
- описание способа обмена данными между АЦП 1273ПВ19Т и микроконтроллером 1986ВЕ1Т на основе метода временного мультиплексирования;
- пример программы обработки данных.

АЦП 1273ПВ19Т

Микросхема 1273ПВ19Т имеет шесть независимых сигма-дельта аналого-цифровых преобразователей, каждый из них оснащён формирователем входного сигнала и усилителем с программируемым коэффициентом усиления [1]. Так же микросхема оснащена внутренним источником опорного напряжения, уровень напряжения которого может быть запрограммирован, и поддерживает соединение до восьми устройств в каскад.

Программирование микросхемы 1273ПВ19Т осуществляется записью данных в восемь восьмибитных регистров управления. После подачи питания АЦП входит в режим управления и ожидает передачи данных. Данные передаются в формате шестнадцатитбитных слов, старшие восемь бит из которых содержат управляющую информацию и адреса регистра и АЦП

в каскаде, младшие восемь – информацию, записываемую в регистр управления.

Функции регистров управления микросхемой следующие. Первый регистр управления определяет режим работы микросхемы АЦП, количество устройств в каскаде, а также позволяет программно сбросить микросхему. Второй регистр отвечает за выбор частот работы АЦП и прореживание. Третий регистр – за подачу питания на все каналы аналого-цифрового преобразователя сразу и за переключение между трёх- и пятивольтовым режимами работы микросхемы. Через четвёртый, пятый и шестой регистры можно управлять каждым каналом АЦП отдельно: настраивать коэффициент усиления и включать, либо отключать питание. Седьмой регистр управления позволяет настроить входы АЦП как однополярные или дифференциальные и сбросить аналоговый модулятор. Восьмой регистр включает инверсный режим у каналов [1].

Протокол обмена данными SSI фирмы Texas Instruments

SSI – Synchronous Serial Interface, это интерфейс с битовой синхронной последовательной передачей данных, данные передаются только в одну сторону. Для передачи данных минимально необходимы четыре провода: один для передачи тактовой частоты, второй – кадровой синхронизации, третий и четвёртый для передачи и приёма данных. Максимальный размер передаваемого сообщения 16 бит.

На рисунке 1 показаны формы сигналов формата SSI:

- 1 – тактовая частота; 2 – кадровая синхронизация;
- 3 – данные; 4 – сигнал выбора чипа.

Считывание каждого бита передаваемой информации происходит по заднему фронту сигнала тактовой частоты (1). Сигнал кадровой синхронизации (2) говорит о начале передачи сообщения.

Обмен данными

Обмен данными между системой обработки цифрового сигнала и АЦП 1273ПВ19Т происходит в формате мультиплексной передачи с временным разделением каналов [5–8]. Каналы выдаются друг за другом с 1-ого по 6-й, затем идёт пауза, и передача повторяется опять. На рисунках 2 и 3 приведён пример такой передачи. Верхний сигнал – это импульсы кадровой синхронизации, а нижний – данные.

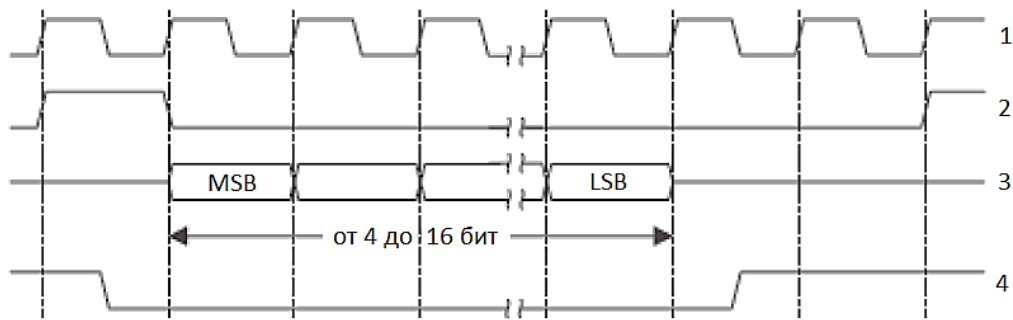


Рис. 1. Формат сигналов SSI



Рис. 2. Передача данных по каналам АЦП 1273PB19T

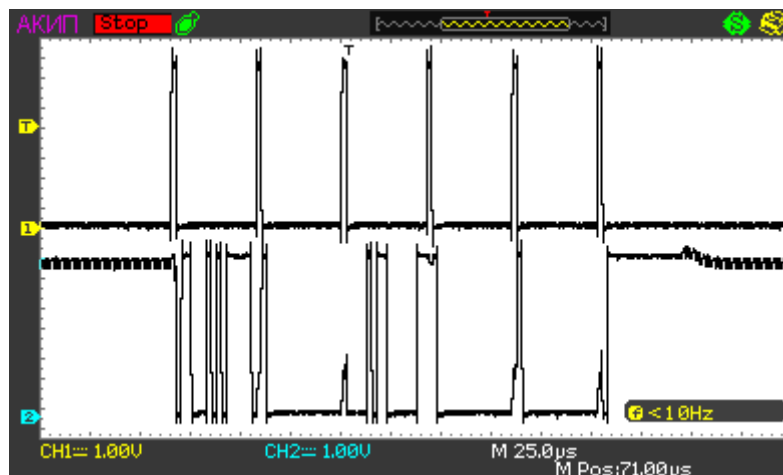


Рис. 3. Кадр передачи данных

Первый и третий каналы передают значение поданного на них напряжения, второй, четвёртый и пятый передают нулевое значение, а шестой канал – максимальное.

Программа обработки данных

В описанном ниже примере АЦП 1273PB19T по очереди передаёт значения напряжения на каждом из шести каналов на самой низкой частоте передачи данных ($f_{DMCLK} / 8$). На шине последовательной передачи данных АЦП является ведущим (контроллером шины), а микроконтроллер 1986BE1T – ведомым. Частота, на которой работает ядро микроконтроллера составляет 128 МГц.

Формат и последовательность сигналов соответствует протоколу, разработанному фирмой Texas In-

struments «Формат синхронного обмена SSI». Вся необходимая настройка аналого-цифрового преобразователя производится микроконтроллером до начала передачи данных, пока настройка не завершена, все данные, принятые с АЦП, игнорируются.

Передачей данных в микроконтроллере управляет таймер 1, прерываниям которого установлен наивысший приоритет. Таймер работает по захвату импульсов кадровой синхронизации с входа ETR1 и по отсчёту времени, необходимому для передачи 16 бит данных. Такая работа таймера обусловлена необходимостью отслеживать номер канала данные с которого передаются в данный момент времени.

Алгоритм обработки сигналов следующий.

Таймер считает пришедшие на его вход импульсы и определяет, с какого канала АЦП приходят данные.

Он измеряет время между приходящими импульсами кадровой синхронизации, при передаче данных время между ними одинаково, после передачи данных с последнего (шестого) канала следует увеличенный интервал в течение которого не происходит передача данных, и, соответственно, не приходят импульсы кадровой синхронизации. Когда таймер определяет, что пришёл увеличенный интервал, он сбрасывает счётчик импульсов. Следующий пришедший на него импульс означает, что идёт передача данных с первого канала АЦП.

При подаче питания или после сброса микросхема АЦП переходит в режим управления. Описываемая в примере программа производит настройку единожды, как только начинают поступать импульсы кадровой синхронизации на вход таймера, затем плата АЦП переводится в режим данных (в котором изменение настроек невозможно) и микроконтроллер только принимает данные.

Порты микроконтроллера 1986BE1T имеют разрядность 16 бит и могут выполнять различные функции. На некоторых из них можно настроить функции входов-выходов каналов контроллера SPI. Подробнее об этом можно почитать в документации на сам микроконтроллер [3].

Для работы шины данных используются пятый, шестой, седьмой и восьмой биты порта C (PC5, 6, 7, 8). Ниже следует код настройки порта:

```
RST_CLK->PER_CLOCK |= (1<<23);
PORTC->OE           |= 0x160;
PORTC->FUNC          |= (0xAA<<10);
PORTC->ANALOG        |= (0xF<<5);
PORTC->PULL          |= 0xFFFF0000;
PORTC->PWR           |= (0xAA<<10);
```

Первой строкой разрешается подача тактовой частоты на все биты порта C, затем происходит настройка бит PC5, PC6, PC7, PC8 как выходов. В третьей строке выполняется назначение альтернативной функции этим битам, в четвёртой – цифровой, далее идёт подтяжка к питанию и формирование импульса с быстрым фронтом.

Следующая часть кода настраивает первый канал контроллера SPI:

```
RST_CLK->PER_CLOCK |= (1<<8);
RST_CLK->SSP_CLOCK |= (1<<24);
SSP1->CR0 |= (1<<4);
SSP1->CR0 |= 0x0F;
SSP1->CR1 |= (1<<2);
SSP1->CR1 |= 0x02;
SSP1->CPSR |= 0x1;
```

```
if ((TIMER1->STATUS & 0x4) == 0x4) {
    if (InitializationInProgress) { // Блок настройки АЦП
        CommandWord++;
        if (CommandWord == 1) { SSP1->DR = 0b1000000010000000; }
        if (CommandWord == 2) { SSP1->DR = 0b1000000000000000; }
        if (CommandWord == 3) { SSP1->DR = 0b1000000101000000; }
        if (CommandWord == 4) { SSP1->DR = 0b1000001001100001; }
        if (CommandWord == 5) { SSP1->DR = 0b1000001100000000; }
        if (CommandWord == 6) { SSP1->DR = 0b1000010000000000; }
        if (CommandWord == 7) { SSP1->DR = 0b1000010100000000; }
        if (CommandWord == 8) { SSP1->DR = 0b1000011010111111; }
        if (CommandWord == 9) { SSP1->DR = 0b1000011100000000; }
        if (CommandWord == 10) { SSP1->DR = 0b1000000000000001; }
```

Первая строка кода разрешает подачу тактовой частоты на контроллер SPI интерфейса, вторая строка включает первый канал SPI. В третьей строке происходит выбор протокола передачи данных, в данном примере – протокол SSI фирмы Texas Instruments. Строкой ниже происходит выбор размер слова данных, который может изменяться в пределах от четырёх до шестнадцати бит, выбран максимальный размер – 16 бит. Далее идёт выбор режима работы передатчика: ведущий либо ведомый, в данном примере выбран ведомый режим. Предпоследняя строка разрешает работу канала, в последней строке, устанавливается делитель частоты.

Настройка таймера так же состоит из двух этапов: настройка нулевого бита порта C как входа ETR1 таймера, и настройка самого таймера на захват импульсов и счёт времени между ними. Поскольку бит порта должен работать как вход, в регистр направления порта записывается ноль.

```
PORTC->OE |= 0;
```

Затем выполняется настройка функции порта согласно руководству на микроконтроллер [3].

```
PORTC->FUNC |= (1<<1);
```

```
PORTC->ANALOG |= 1;
```

Настройка работы самого таймера выглядит следующим образом:

```
RST_CLK->PER_CLOCK |= (1<<14);
RST_CLK->TIM_CLOCK |= (1<<24);
TIMER1->CNTRL = 0;
TIMER1->CNT = 0;
TIMER1->PSG |= 8;
TIMER1->ARR = 2000;
TIMER1->CNTRL |= 1;
NVIC->IP[14] |= 0b00000000;
TIMER1->IE |= (1<<2);
TIMER1->IE |= (1<<1);
NVIC_EnableIRQ(TIMER1_IRQn);
```

Первые две строки подают тактовую частоту на блок таймера 1 и включают приём этой частоты, далее идут сброс настроек и счётчика таймера, пятая строка устанавливает делитель тактовой частоты счётчика, шестая – устанавливает число, до которого должен досчитать таймер, а седьмая – его запускает. Четыре последние строки настраивают прерывания таймера по захвату и по счёту, давая им наивысший приоритет.

Обработчик прерываний таймера представляет из себя функцию «void Timer1_IRQHandler(void)», размещаемую в одном файле с основной функцией «main». В теле функции обработки прерывания по таймеру 1 размещён следующий код:

```

if(CommandWord == 1) { SSP1->DR = 0x7FFF
InitializationInProgress = 0;
CommandWord = 0; }
}
else if( !InitializationInProgress ) { // Блок сбора данных
Word++;
    if(Word == 1) { DAC5 = SSP1->DR; }
    else if(Word == 2) { DAC6 = SSP1->DR; }
    else if(Word == 3) { DAC1 = SSP1->DR; }
    else if(Word == 4) { DAC2 = SSP1->DR; }
    else if(Word == 5) { DAC3 = SSP1->DR; }
    else if(Word == 6) { DAC4 = SSP1->DR; }
}
TIMER1->CNT = 0;
TIMER1->STATUS &= ~(1<<1);
TIMER1->STATUS &= ~(1<<2);
}
if(((TIMER1->STATUS & 0x2) == 0x2)) {
    while((SSP1->SR & (1<<3)) == (1<<3)) { eraserOfDataFromSPI1273 = SSP1->DR; }
    TIMER1->STATUS &= ~(1<<1); Word = 0;
}
}

```

Логически обработчик прерывания делится на три блока: блок настройки микросхемы АЦП, блок сбора данных и блок сброса по счётчику таймера, последний блок необходим для определения паузы, возникающей после передачи значений всех шести каналов.

Блок настройки состоит из одиннадцати командных слов и проверки выполнения условия работы АЦП в режиме управления. Первое командное слово сбрасывает микросхему АЦП. Второе командное слово устанавливает режим управления. Третье – настраивает тактовую частоту работы. Четвёртое слово включает питание на всех шести каналах и переводит микросхему в режим работы от источника напряжения 3,3 вольта. Пятое, шестое и седьмое командные слова отключают усиление на всех каналах АЦП. Восьмое командное слово переводит все входы АЦП в режим однополярного входа. Девятое – включает нормальный (неинверсный) режим работы входов АЦП. Десятое слово переводит микросхему в режим выдачи данных. Заключительное, одиннадцатое слово, представляет собой пустую команду, которая должна выдаваться во время работы микросхемы, в этот момент также происходит переход микроконтроллера 1986BE1T в режим приёма данных.

Блок сбора данных считывает данные при каждом новом импульсе кадровой синхронизации, пришедшем на вход таймера. Приём значения первого канала сдвинут на два шага в силу особенностей контроллера шины синхронного обмена данных, когда приходит первый импульс, контроллер шины начинает заполнение буфера приёмника, за время второго

кадрового импульса данные из буфера приёмника переписываются в регистр, из которого ядро микроконтроллера 1986 может извлечь их и обработать.

Третий блок срабатывает по истечению времени ожидания следующего импульса, очищает буфер приёмника SPI и устанавливает счётчик на приём значения первого канала АЦП.

Приведенный код программы используют множественные условия «if» и «if-else» для наглядности отображения командных слов, здесь несомненно можно использовать более эффективные способы записи приведенного кода, например, с использованием конструкции «switch-case» [9], но авторы статьи акцент сделали на наглядности записи командных слов.

Вывод

В статье описан алгоритм управления и сбора данных с аналого-цифрового преобразователя 1273ПВ19Т посредством последовательной передачи с временным мультиплексным разделением каналов. Для отслеживания каждого из каналов был использован первый таймер микроконтроллера 1986BE1T, настроенный на работу по счёту и по захвату. Такое решение было принято в связи с ограничениями контроля заполнения буфера модуля SPI микроконтроллера, что затрудняло определение номера принимаемого канала АЦП.

В статью не вошло рассмотрение всех возможных режимов работы 1273ПВ19Т, передачи данных по последовательному интерфейсу на более высокой скорости, управление каналами и соединение нескольких АЦП каскадом, что требует отдельного описания.

Литература:

1. Микросхема интегральная 1273ПВ19Т. Руководство пользователя. [Электронный ресурс] URL: <https://eandc.ru/pdf/mikroskhema/1273pv19t.pdf> (дата обращения: 28.10.2020).
2. AD73360 Datasheet (PDF) – Analog Devices. [Электронный ресурс] URL: <https://pdf1.alldatasheet.com/datasheet-pdf/view/48191/AD/AD73360.html> (дата обращения: 28.10.2020).
3. Спецификация на микросхему 1986BE1T. [Электронный ресурс] // Официальный сайт АО «ПКК Ми-ландр». URL: https://ic.milandr.ru/products/mikrokontrollery_i_protessory/32_razryadnye_mikrokontrollery (дата обращения: 28.10.2020).
4. Ключева С.Ф. Организация передачи данных для микроконтроллера 1986BV1T // Евразийское научное объединение. 2017. Т. 1. № 8(30). С. 28-30.

5. Jehad M. Hamamreh, Abdulwahab Hajar, Mohamedou Abewa. Orthogonal Frequency Division Multiplexing With Subcarrier Power Modulation for Doubling the Spectral Efficiency of 6G and Beyond Networks. //Transaction on Emerging Telecommunications Technologies. 31(4), e3921.
6. Miss. Sneha Kumari, Mr Ankit Tripathi. BER Performance of OFDM System in Rayleigh Fading Channel Using Cyclic Prefix International // Journal of Advanced Engineering Research and Science. Vol. 4. Issue -11. Nov- 2017.
7. Gilad Katz, D.Sadot. Time-Division Multilevel Multiplexing Communication Method // IEEE Photonics Technology Letters. Vol. 19. No 20. 2007.
8. Крюков Я.В. Демидов А.Я., Покаместов Д.А. Метод множественного доступа с разделением каналов по мощности на ортогональных несущих // Т-Comm: Телекоммуникации и транспорт. 2018. Том 12. № 1. С. 17-22.
9. О. Ю. Рязанов, Ю. Д. Рязанов. Функциональная реализация конструкции switch-case в языке java // Вестник ВГУ. Серия: Системный анализ и информационные технологии. 2017. № 4. С. 114-116.