МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Нижегородский государственный университет им. Н.И. Лобачевского
Национальный исследовательский университет

РАБОТА С ЦАП И АЦП МИКРОКОНТРОЛЛЕРА СЕРИИ MSP430

Практикум

Рекомендовано методической комиссией радиофизического факультета для студентов ННГУ, обучающихся по направлениям подготовки 03.03.03 «Радиофизика», 02.03.02 «Фундаментальная информатика и информационные технологии» и специальности 10.05.02 «Информационная безопасность телекоммуникационных систем»

Нижний Новгород
2015

УДК 681.3

ББК 32.973.2

РАБОТА С ЦАП И АЦП МИКРОКОНТРОЛЛЕРА СЕРИИ MSP430 / Составители: Шкелёв Е.И., Пархачёв В.В., Ивлев Д.Н., Семенов В.Ю. – Практикум. – Нижний Новгород: Нижегородский госуниверситет, 2015. – 26с.

Рецензент: доцент кафедры общей физики С.Н. Жуков

Методические указания содержат описание настроек аналогово-цифрового и цифро-аналогового преобразователей, входящих в контроллер MSP430F2618. На примере элементарных программ на языке ассемблера разобран вариант настройки регистров микроконтроллера, отвечающих за работу с АЦП и ЦАП.

В общем виде даётся описание структуры программы для контроллера MSP430 на языке высокого уровня Си. Представлены фрагменты программ, управляющих режимами функционирования периферийного оборудования контроллера MSP430. Даны необходимые пояснения по обработке оцифрованного сигнала на примере фильтрации нижних частот методом скользящего среднего. Приводятся задания для самостоятельного выполнения. Имеется перечень контрольных вопросов.

Методические указания к лабораторной работе разработаны для преподавания на радиофизическом факультете дисциплины «Аппаратные средства вычислительной техники» по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии», дисциплины «Микропроцессорные системы» по направлению подготовки 03.03.03 «Радиофизика» и дисциплины «Аппаратные средства телекоммуникационных систем» по специальности 10.05.02 «Информационная безопасность телекоммуникационных систем».

Ответственные за выпуск:
председатель методической комиссии радиофизического факультета ННГУ, к.ф.-м.н., доцент **Н.Д. Миловский**,
зам. председателя методической комиссии радиофизического факультета ННГУ, д.ф.-м.н., профессор **Е.З. Грибова**

УДК 681.3
ББК 32.973.2

1. Введение

Цель лабораторной работы состоит в развитии навыков составления и отладки программ для микроконтроллеров (МК) фирмы Texas Instruments серии MSP430 [1-6]. В частности, рассматривается управление такими периферийными модулями, как АЦП и ЦАП.

Лабораторная работа является продолжением работ "Знакомство с микроконтроллером серии MSP430" и "Первые шаги в программировании микроконтроллера серии MSP430".

2. Модуль 12 битного АЦП [4, 5]

Модуль ADC12 обеспечивает быстрое выполнение 12-битных аналого-цифровых преобразований. В составе модуля имеется 12-битное ядро с регистром последовательного приближения, блок управления выборкой, генератор опорного напряжения и буфер (conversion memory) размером 16 слов. Этот буфер позволяет модулю формировать до 16 независимых выборок с последующим преобразованием и сохранением результата без использования ЦПУ.

Модуль ADC12 имеет следующие особенности:

- максимальная скорость преобразования более 200 тыс. выборок/с;
- 12-битный преобразователь с монотонной характеристикой без пропуска кодов;
- устройство выборки/хранения с программируемым временем выборки, определяемым таймерами или программно;
- запуск преобразования производится программно, по сигналу от Таймера А или по сигналу от Таймера В;
- программно конфигурируемый внутренний генератор опорного напряжения (1.5 В или 2.5 В);
- внешний или внутренний источник опорного напряжения (выбирается программно) ;
- восемь каналов преобразования внешних сигналов;

– каналы преобразования для внутреннего датчика температуры, напряжения AVCC и внешних опорных напряжений;
– независимые источники положительного и отрицательного опорного напряжений, задаваемые отдельно для каждого канала;
– конфигурируемый источник тактового сигнала;
– четыре режима преобразования: одноканальный, многократный одноканальный, последовательный и многократный последовательный;
– ядро АЦП и источник опорного напряжения могут выключаться независимо друг от друга;
– регистр вектора прерывания для быстрого декодирования 18 прерываний АЦП;
– 16 регистров для хранения результатов преобразований.

Конфигурирование модуля ADC12 осуществляется пользовательской программой. Настройка его работы производится посредством выставления соответствующих битов регистров ADC12CTL0 и ADC12CTL1. Также предусмотрено 16 регистров ADC12MEMx для хранения результатов преобразований. Конфигурация каждого из регистров ADC12MEMx задаётся соответствующим регистром управления ADC12MCTLx. Подробное описание содержимого регистров модуля ADC12 содержится в руководстве [4, 5].

Физически входы восьми каналов модуля АЦП контроллера MSP430F2618 могут быть соединены с восемью ножками порта ввода/вывода P6 (ножки P6.0 – P6.7) посредством соответствующей настройки порта P6.

3. Модуль 12 битного ЦАП [4, 5]

Модуль DAC12 представляет собой 12-битный ЦАП с потенциальным выходом. Модуль DAC12 может работать в 8- или 12-битном режимах и использоваться совместно с контроллером прямого доступа к памяти DMA. При наличии в устройстве нескольких модулей, они могут быть сгруппированы для синхронного выполнения операции обновления.

Модуль DAC12 имеет следующие особенности:

- монотонная характеристика в пределах 12-битного диапазона;
- - потенциальный выход с разрешением 8 или 12 бит;
- программируемое соотношение между временем установления и потреблением;
- внешний или внутренний источник опорного напряжения;
- использует данные в натуральном двоичном коде или в дополнительном коде;
- возможность самокалибровки для коррекции напряжения смещения;
- возможность синхронного обновления выходов нескольких модулей DAC12.

Конфигурирование модуля DAC12 осуществляется пользовательской программой. Настройка его работы производится посредством выставления соответствующих битов регистров DAC12_xCTL. Если планируется использовать внутренний источник опорного напряжения, то следует сконфигурировать регистр ADC12CTL0. Данные для оцифровки следует помещать в регистры DAC12_xDAT. Подробное описание содержимого регистров модуля DAC12 содержится в руководстве [4, 5].

Физически два выходных канала модуля ЦАП контроллера MSP430F2618 могут выведены контакты P6.6 и P6.7 порта.

4. Описание установки

Для выполнения лабораторной работы используются:

- - отладочная плата OLIMEX MSP430-P2618 [7] с установленным микроконтроллером MSP430F2618 (рис. 1);
- - генератор электрических сигналов произвольной формы;
- - универсальный осциллограф;

– - комплект кабелей с разъёмом BNC на одном конце (для подключения к генератору и осциллографу), и двумя разъёмами на другом конце для подключения к штырьковым разъёмам отладочной платы.

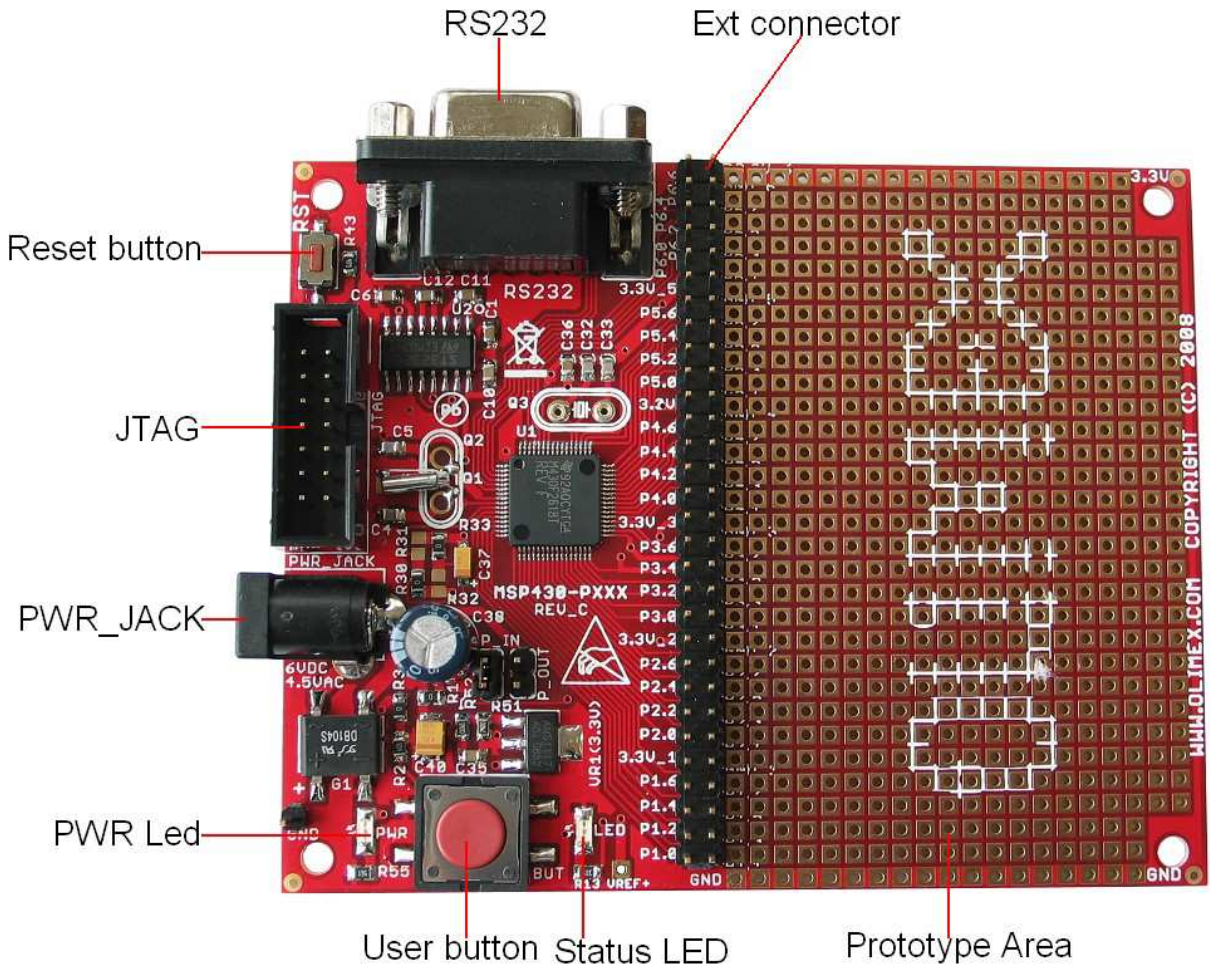

Рис. 1. Отладочная плата

ВНИМАНИЕ! При работе с микроконтроллером во избежание поломок необходимо соблюдать осторожность:

1) запрещается трогать руками штырьковые контакты отладочной платы, поскольку контроллер может быть повреждён статическим электричеством;

2) узнайте схему подключения кабелей к отладочной плате у преподавателя, прежде чем соединять их;

Pin #	Signal Name	Pin #	Signal Name
1	VCC	2	GND
3	P6.6	4	P6.7
5	P6.4	6	P6.5
7	P6.2	8	P6.3
9	P6.0	10	P6.1
11	VCC	12	GND
13	P5.6	14	P5.7
15	P5.4	16	P5.5
17	P5.2	18	P5.3
19	P5.0	20	P5.1
21	VCC	22	GND
23	P4.6	24	P4.7
25	P4.4	26	P4.5
27	P4.2	28	P4.3
29	P4.0	30	P4.1
31	VCC	32	GND
33	P3.6	34	P3.7
35	P3.4/TXD0	36	P3.5/RXD0
37	P3.2	38	P3.3
39	P3.0	40	P3.1
41	VCC	42	GND
43	P2.6	44	P2.7
45	P2.4	46	P2.5
47	P2.2	48	P2.3
49	P2.0	50	P2.1
51	VCC	52	GND
53	P1.6	54	P1.7
55	P1.4	56	P1.5
57	P1.2	58	P1.3
59	P1.0	60	P1.1

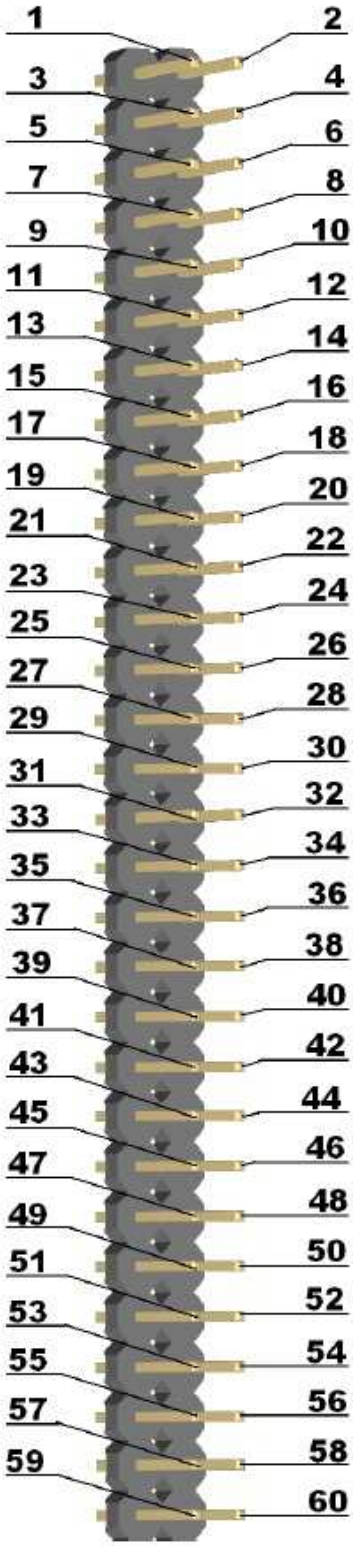

Рис. 2. Разъёмы на отладочной плате

3) категорически запрещается подавать на вход АЦП сигнал амплитудой больше 2.5 Вольт, то есть следует внимательно следить за тем, чтобы

на выходе генератора значение напряжения не превышало 2.5 Вольт. Также не следует подавать отрицательное напряжение.

На плате имеется 60-штырьковый разъём (pin connectors). Обратите внимание на подписи около каждого из контактов. Для удобства эти подписи продублированы на рис. 2. Легко видеть, что кроме портов P1 – P6 на разъём несколько раз выведены напряжение питания (VCC = +3.3 В) и "земля" (GND).

Сигнал на АЦП подаётся с генератора, с ЦАП выводится на осциллограф. Для этого используются дополнительные кабельные соединения и провода. "Землю" генератора и осциллографа (провода, помеченные чёрным цветом) необходимо подключить к любым выбранным для этого контактам GND. Сигнал от генератора (кабель с красным проводом) следует подключить к выводу порта P6.2 (уточните у преподавателя). Сигнал на осциллограф выводится также через кабель с красным проводом осциллографа, и его следует подключить к выводу порта P6.6 (уточните у преподавателя).

5. Настройка работы АЦП и ЦАП.

5.1. Пример программы на языке ассемблера

Рассмотрим простейшую программу на языке низкого уровня, которая одновременно задействует АЦП и ЦАП. Назначение этой программы чрезмерно простое – оцифровывать аналоговый сигнал, поданный на вход АЦП, и передавать его без изменений на выход ЦАП. Получить код программы на свой лабораторный компьютер можно у преподавателя (проект "adc_dac"). "Шапка" программы была подробно описана в указаниях к работе "Первые шаги в программировании микроконтроллера серии MSP430". Прочие комментарии и весь текст программы приводятся в Листинге 1.

Листинг 1

```
#include "msp430x26x.h"
            NAME    main                    ; module name
            PUBLIC  main                    ; make the main label
                                    ; vissible outside this module
            RSEG    CSTACK                  ; Define stack segment
            RSEG    CODE                    ; Assemble to Flash memory
;-------------------------------------------------------------------
main
RESET       mov.w   #SFE(CSTACK),SP
StopWDT     mov.w   #WDTPW+WDTHOLD,&WDTCTL    ; Остановка WDT
;-------------------------------------------------------------------
SetupADC12  mov.w   #SHT0_2+ADC12ON+REFON+REF2_5V,&ADC12CTL0
          ; включаем модуль ADC12, используем внутренний источник
          ; опорного напряжения 2.5 В
            mov.w   #SHP+CSTARTADD_0,&ADC12CTL1 ; выборка по
          ; таймеру выборки, а результат сохраняется в ADC12MEM0

            mov.b   #INCH1,&ADC12MCTL0        ; выбран входной
          ; канал A2 (входной сигнал берём с ножки P6.2)
            bis.w   #ENC,&ADC12CTL0           ; разрешение работы
          ; АЦП. Бит должен выставляться после остальной
          ; настройки (последним)
;-------------------------------------------------------------------
SetupDAC12  mov.w   #DAC12IR+DAC12AMP_5+DAC12ENC,&DAC12_0CTL
            ;DAC12 включить, работает на ножку P6.6
;-------------------------------------------------------------------
TimerA      MOV     #TASSEL0+TACLR, &TACTL  ;Настройка таймера A
            BIS     #CCIE,&CCTL0
            MOV     #10h,&CCR0              ;Период таймера A
            BIS     #MC0, &TACTL
            EINT
;-------------------------------------------------------------------
```

```
            bis.b   #BIT2,&P6SEL              ; указываем, что
          ; ножка P6.2 будет использоваться
          ; для периферийного модуля (ADC)
Mainloop
            jmp     Mainloop                 ; пустой цикл
TimerA_ISR
            bis.w   #ADC12SC,&ADC12CTL0
          ; запускаем однократное преобразование
          ; конструкция устанавливает бит0 = 1 в регистре
          ; ADC12CTL0, те запускает преобразование АЦП
loop        MOV     &ADC12CTL1, R14
            AND     #01h,R14
          ;ожидаем завершения преобразования
            JNZ     loop
            MOV.w   &ADC12MEM0,R15
            mov.w   R15,&DAC12_0DAT
            reti
            COMMON  INTVEC                   ; Interrupt Vectors

            ORG     TIMERA0_VECTOR           ; Timer_A0 Vector
            DW      TimerA_ISR

            ORG     RESET_VECTOR             ; POR, ext. Reset
            DW      RESET
            END
```

Частота дискретизации АЦП определяется Таймером А. и оцифрованные данные принимаются микроконтроллером по запросам прерывания от этого таймера. При обработке запросов прерывания от таймера используется предварительно настроенный для этого однократный режим запуска АЦП с выбранным для приёма сигнала от генератора канлом.

Задания:

1) Соберите измерительную установку, подключив генератор и осциллограф к отладочной плате согласно инструкциям преподавателя и замечаниям из раздела 4.

2) Установите на генераторе режим генерации прямоугольных импульсов и настройте сигнал в виде меандра с амплитудой 0.5 В и частотой 100 Гц (как показано на рис. 3). При этом нижнее значение напряжения необходимо сделать равным нулю, а верхнее 0.5 В. Пронаблюдайте сигнал на осциллографе.

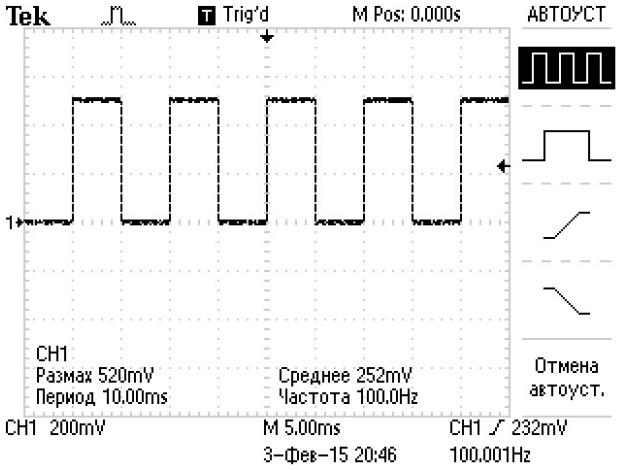

Рис. 3. Пример входного сигнала

3) Какая частота дискретизации входного сигнала обеспечивается программой из листинга 1 по умолчанию? Измените частоту дискретизации в несколько раз как в большую, так и в меньшую сторону. Пронаблюдайте эффект такого изменения. При какой частоте дискретизации сигнал на выходе контроллера (на экране осциллографа) изменил форму? Объясните результат.

4) Измените программу таким образом, чтобы входной аналоговый сигнал с генератора можно было подавать на ножку P6.3. Какие регистры контроллера и каким образом пришлось изменить?

5.2. **Генератор треугольных импульсов**

Рассмотрим пример программы, позволяющей использовать микроконтроллер как генератор сигналов произвольной формы (листинг 2). Её код в электронном виде можно получить у преподавателя (проект "dac_pila").

Листинг 2

```
;  *                  * ^
; **                *  * |
;* *              *    * |
;  *            *      * |
;  *         *         * | 2.44 v
;  *      *            * |
;  *    * <->          * |
;  **     1ms          * |
;   <-----------------> v
;           40ms

#include "msp430x26x.h"
            NAME    main                    ; module name
            PUBLIC  main                    ; make the main label
                                            ; vissible
                                            ; outside this module
            RSEG    CSTACK                  ; Define stack segment
            RSEG    CODE                    ; Assemble to Flash
main
RESET       mov.w   #SFE(CSTACK),SP           ;
StopWDT     mov.w   #WDTPW+WDTHOLD,&WDTCTL    ; Остановка WDT
;-------------------------------------------------------------
            mov.w   #REF2_5V+REFON,&ADC12CTL0
            ;настраиваем генератор опорного напряжения
SetupDAC12  mov.w   #DAC12IR+DAC12AMP_5+DAC12ENC,&DAC12_0CTL
            ;DAC12 включить, работает на ножку P6.6
;-------------------------------------------------
```

```
TimerA      MOV     #TASSEL0+TACLR, &TACTL   ; настройка таймера A
            BIS     #CCIE,&CCTL0
            MOV     #0021h,&CCR0             ; период таймера A
            BIS     #MC0, &TACTL
;-------------------------------------------------------------------
            bis.b   #BIT0,&P6DIR
            mov.b   #000h,&P6OUT             ; зажигаем светодиод

            MOV     #0FA0h,R4                ;амплитуда пилы
            mov.b   #0h,R15
            EINT

Mainloop
            jmp     Mainloop
TimerA_ISR
            Add     #0064h,R15          ; увеличиваем R15 на сто
            CMP     R4,R15              ; ?R15<=4000
            JN      OUTPUT
            mov.w   #0h,R15
            XOR.b   #BIT0,&P6OUT        ; инвертируем светодиод
OUTPUT      mov.w   R15,&DAC12_0DAT
            reti
;-------------------------------------------------------------------
            COMMON  INTVEC                  ; Interrupt Vectors
            ORG     TIMERA0_VECTOR          ; Timer_A0 Vector
            DW      TimerA_ISR
            ORG     RESET_VECTOR            ; POR, ext. Reset
            DW      RESET
            END
```

Задания:

1) Отключите от входа АЦП внешний генератор.

2) Изучите текст программы? Что она делает?

3) Запустите её, пронаблюдайте результат на осциллографе. Совпадает ли он с данными рис. 4.

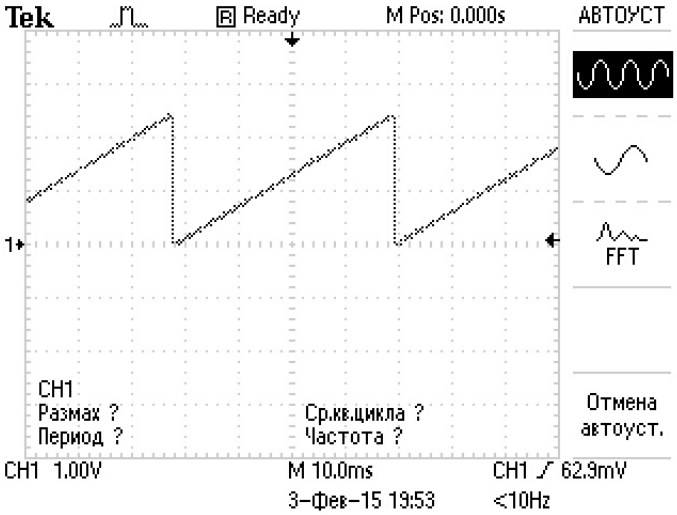

Рис. 4. Треугольные импульсы

4) Какие параметры программы задают частоту дискретизации, период следования импульсов и их амплитуду?

5) "Приблизьте" сигнал на экране осциллографа с помощью ручек "развёртка" и "амплитуда". Рассмотрите изображение. Является ли нарастающий фронт треугольных импульсов идеально прямым? Объясните наблюдения.

6) Измените программу таким образом, чтобы она генерировала импульсы в виде равнобедренных треугольников.

7) Измените программу таким образом, чтобы она генерировала импульсы в виде прямоугольных импульсов.

8) Измените программу таким образом, чтобы она генерировала синусоидальное напряжение.

6. Программирование АЦП и ЦАП на языке высокого уровня

6.1. Общая структура программы на языке Си

Для того, чтобы научиться писать программы для контроллера MSP430 прежде всего необходимо знать структуру основной программы *main( )* или, как ещё говорят, структуру основного цикла исполнения программы.

В листинге 3 приведена структура программы *main( ),* а также функция обработки прерывания от некого устройства периферии.

Листинг 3

```
//=========================================================
#include <msp430x26x.h> // подключение заголовочного файла с
                        //основными константами и регистрами
                        //конкретного контроллера

#define   Num_of_Results   32 // определяем макроподстановку

volatile double koef = 0.03125;// определяем переменную
                               // плавающего типа, которую
                               // можно изменить извне,

// тело основной программы
int main(void)
{
   //здесь программируются режимы работы элементов периферии,
   //например, таймеры, АЦП, ЦАП т.д.

   while (1) // определяем бесконечный цикл
   {
```

```
        //  здесь описывается тело цикла (то, что должна
        //  выполнить программа)
    } // конец бесконечного цикла

}// конец основной программы

// функция обработки прерывания,  например от таймера А
#pragma vector = TIMERA0_VECTOR // обязательная строка при
                                //описании прерывания от
                                //таймера А, например
__interrupt void TA0_ISR(void) // заголовок функции обработки
                               //прерывания
{
   //тело функции обработки прерывания (то, что будет
   //выполнять контроллер при возникновении прерывания)
}
//=======================================================
```

При программировании важно помнить несколько моментов. Во-первых, названия всех регистров контроллера и векторов прерываний находятся в файле msp430x26x.h.

Во-вторых, необходимо отконфигурировать (настроить) все периферийные устройств из числа тех, которые будут использоваться в целевой системе. Это необходимо сделать перед тем, как к этим устройствам будет обращение со стороны программы. Хотя по умолчанию все элементы периферии уже находятся в каком-то режиме. Уточнить настройки по умолчанию можно в [6].

В-третьих, программа исполняется в бесконечном цикле до тех пор, пока не происходит прерывание от элемента периферии. Такой режим работы предполагает установку в процессе инициализации системы глобального бита разрешения прерываний в регистре состояния ядра микроконтроллера. При жела-

ции пользуясь этим битом программист вообще может запретить восприятие ядром запросов прерывания. Точно также можно запретить работу в режиме прерываний отдельных устройств ввода/вывода посредством соответствующей настройки.

В-четвертых, название функции прерывания должно начинаться с «`__interrupt`», это служебное слово обычно используется в синтаксисе языка Си для указания того, что соответствующий программный модуль относится к процедуре обработки прерывания. Сочетание слов является «`#pragma vector = TIMERA0_VECTOR`» аппаратно ориентированным и показывает на элемент таблицы прерываний контроллеров семейства MSP430 (на вектор прерываний).

Всё сказанное выше относится к тому случаю, когда тело основной программы обработки данных от АЦП (тело бесконечного цикла, в частности), а также тело функции обработки прерывания пишется на стандартном языке ANSI C.

6.2. Программирование аналого-цифрового преобразователя (АЦП)

Настройку режима работы аналого-цифрового преобразователя (АЦП) можно разделить на две части: физическую и программную. С физической точки зрения режим работы АЦП определяется, прежде всего, частотой дискретизации входного сигнала, входной аналоговой полосой частот пропускания, разрядностью цифрового кода, величиной первой побочной гармоники, а также максимальным значением напряжения, которое он может принять на свой вход. Для контролера входящего в лабораторную установку максимальный уровень входного напряжения составляет 2.5 В.

ВНИМАНИЕ! Категорически запрещается подавать на вход АЦП сигнал амплитудой больше 2.5 Вольт! Входом для АЦП могут служить 8 но-

жек контроллера P6.0 – P6.7. Рекомендуется при выполнение работы использовать ножку P6.2, поскольку к ножкам P6.0 и P6.1 на отладочной плате подключены светодиод и кнопка.

Частота дискретизации АЦП задается различными способами. Одним из наиболее распространенных является использование таймеров, в частности Таймера А или сторожевого таймера (WDT). Выборка аналогового сигнала и его преобразование в двоичный цифровой код начинается при каждом циклически повторяющемся обнулении заданного при программировании двоичного кода. Тактовая частота, с которой работает таймер контроллера в лабораторной установке равна 32 кГц.

Для того, чтобы научиться программно настраивать АЦП, изучите листинг 4 и пояснения в нём.

Листинг 4

```
//=========================================================
WDTCTL = WDT_MDLY_0_064; // задаем период прерывания
                         //сторожевого таймера через 0.064 мс
IE1 |= WDTIE; // разрешаем прерывание от сторожевого таймера

TACCR0 = 13600; // задаем значение, до которого будет считать
                //таймер А
TACCTL0 |= CCIE; // разрешаем порывание от таймера А при
                 //достижении значения TACCR0
TACCTL0 &= ~CCIE; // запрещаем порывание от таймера А
TACTL = TACLR + MC_1 + TASSEL_2; // команда для однократного
                                 //обнуления таймера,
                                 //установки направления
                                 //счета вверх, выбор
                                 //источника тактирования

P6SEL    |= BIT2; // вход P6.2 будет использоваться
                  // периферийным модулем (АЦП)
```

```
ADC12MCTL0 |= INCH1;  // сообщаем АЦП, что входом для него
                      // будет ножка P6.2
ADC12CTL0 = REF2_5V + REFON; // задаем диапазон напряжения
                             //для оцифровки АЦП
ADC12CTL0 = ADC12ON+SHT0_2;  // включаем АЦП и задаем число
                             //тактов для преобразования
                             //аналогового сигнала в цифровое
                             //значение
ADC12CTL1 = SHP; // указываем, что в качестве источника
                 //частоты дискретизации будет таймер
ADC12CTL0 |= ENC;// преобразование в цифру разрешено

ADC12CTL0 |= ADC12SC;// программный запуск АЦП в нужном месте
                     //программы
//=========================================================
```

Фрагменты листинга 4 не описывают все настройки АЦП и таймера, однако они могут быть полезными при выполнении заданий лабораторной работы. Необходимо отметить, что цифровой код от ножки P6.2 хранится в регистре ADC12MEM0. А в регистре ADC12IFG хранятся флаги возникновения прерывания для всех каналов АЦП. Как только текущее преобразование выполнено, флаг прерывания (один из битов в регистре ADC12IFG) сбрасывается. Брать следующий отсчёт и делать оцифровку можно только после сброса установленного ранее флага запроса прерывания.

6.3. **Программирование цифро-аналогового преобразователя (ЦАП)**

Настройку режима работы цифро-аналогового преобразователя (ЦАП) также можно разделить на две части: физическую и программную. С точки зре-

ния физики важно понимать, что у ЦАП есть максимальное выходное напряжение, которое он способен выдать, выходное сопротивление, максимальная частота преобразования цифрового кода в аналоговый сигнал, выходная аналоговая полоса частот пропускания, а также разрядность цифрового кода.

С программной точки зрения при настройке режима работы ЦАП необходимо указать контроллеру следующие параметры. Во-первых, необходимо указать ножку контролера, на которую будет подан выходной аналоговый сигнал с ЦАП, и настроить режим ее работы. В контроллере для данной лабораторной работы это может быть либо ножка P6.6 для ЦАП №0, либо ножка P6.7 для ЦАП №1. Это определено архитектурой используемого контроллера.

Во-вторых, контроллеру необходимо указать, какое максимальное напряжение будет выдавать ЦАП. Это может быть либо 2.5 В, либо 1.5 В.

В-третьих, необходимо указать контроллеру источник сигнала, запускающего преобразование цифрового кода в аналоговый сигнал. Это может быть прерывание от таймера А, прерывание от таймера В, а также простое перемещение цифрового кода в регистр DAC12_0DAT или DAC12_1DAT (как только происходит присвоение значения этому регистру начинается новое преобразование вне зависимости от выполнения результатов предыдущего преобразования).

Для того, чтобы научиться программно настраивать ЦАП изучите листинг 5 и пояснения в нём.

Листинг 5

```
//=========================================================
ADC12CTL0 = REF2_5V + REFON; // установили максимальное
                             //напряжение выхода ЦАП в 2,5 В

DAC12_0CTL = DAC12IR + DAC12AMP_5 + DAC12ENC; // установили

// множитель выходного напряжения равным 1
// установили среднюю скорость преобразования
//включили модуль ЦАП
```

```
DAC12_0CTL = DAC12LSEL_0; // указали в качестве сигнала
                          //запуска загрузки защелки ЦАП
                          //загрузку в регистр DAC12_0DAT
//=======================================================
```

При программировании необходимо помнить, что цифровой код выходного сигнала ЦАП необходимо помещать в регистр DAC12_0DAT или DAC12_1DAT.

6.4. Цифровая фильтрация нижних частот методом скользящего среднего

Произвести цифровую фильтрацию нижних частот сигнала, поданного в контроллер (на АЦП) с генератора сигналов предлагается методом «окна скользящего среднего». Принцип работы данного метода заключается в следующем. Во-первых, выбирается размер окна, в котором будет производиться усреднение, допустим 32 отсчета сигнала. Этот размер влияет на амплитудно-частотную характеристику (АЧХ) фильтра. Построить АЧХ можно, например, в пакете MATLAB с помощью функции freqz() при надлежащем выборе входных параметров.

Во-вторых, среди первых 32 отсчетов входящего в фильтр сигнала находится арифметическое среднее. Это первый отсчет выходного сигнала после фильтра.

В-третьих, сдвигаем окно на один отсчет и повторяем процедуру нахождения арифметического среднего и т.д.

Для того, чтобы научиться программировать фильтр «окна скользящего среднего» изучите листинг 6 и пояснения в нем.

Листинг 6

```
//=======================================================
#define   Num_of_Results  32 // задаем размер окна усреднения

volatile unsigned int results[Num_of_Results]; // массив для
вычисления арифметического среднего
volatile double mean_sig = 0.0; // переменная для среднего
volatile double norm_koef = 0.03125;//нормирующий коэффициент

unsigned int ii = 0; // переменные для циклов
unsigned int jj = 0;



results[ii] = ADC12MEM0; // заносим в массив очередное значе-
ние сигнала с АЦП

if (ii <Num_of_Results) // организуем циклическую запись в
                        //массив
   ii = ii + 1;
else
   ii = 0;

for (jj = 0;jj<Num_of_Results;jj++) // находим арифметическое
                                    //средее
{
   mean_sig = mean_sig + results[jj];

}

mean_sig = norm_koef*mean_sig;
//=======================================================
```

Полное описание устройства и регистров таймеров, АЦП, ЦАП, сторожевого таймера можно найти по ссылке [8]. Обширную информацию (включая ба-

зовые примеры программ) по контроллеру можно найти на сайте официального производителя [9].

***Задания*:**

1) Получите у преподавателя программу (проект "filter"), которая позволит произвести цифровую фильтрацию нижних частот сигнала, поданного на АЦП контроллера с генератора сигналов. Результат фильтрации передаётся на ЦАП и может быть выведен на осциллографа. Подайте с генератора синусоидальный сигнал такой, что минимальное значение напряжения равно нулю, а максимальное 2 В. Изменяя частоту синуса от нуля до 10 кГц, снимите частотную характеристику фильтра. Как изменится эта характеристика, если увеличить/уменьшить вдвое длину окна усреднения? Постройте графики измеренных характеристик.

2) Напишите программу на языке Си, которая позволит снять передаточную характеристику АЦП. Предполагается, что с генератора сигналов по очереди подаются напряжения U в диапазоне [0..2,5] Вольт. В режиме отладчика необходимо посмотреть и записать цифровой код D, вырабатываемый АЦП, на каждый из поданных сигналов. Полученную зависимость необходимо построить на графике $U(D)$.

3) Напишите программу на языке Си, которая позволяет сгенерировать в контроллере синусоидальный сигнал, частота этого сигнала задается преподавателем. Предполагается, что при проверке программы к плате с контроллером подсоединяется осциллограф, на экране которого должен отобразиться синусоидальный сигнал заданной частоты.

Пишите свои программы!

7. Контрольные вопросы

1. Объясните структуру и логику исполнения в контроллере основной программы main() на языке ассемблера и Си.
2. Опишите синтаксис и логику исполнения в контроллере функции обработки прерывания от устройства периферии на языке ассемблера и Си.
3. Перечислите основные регистры таймера А объясните способ записи в них управляющего кода при программировании на языках АССЕМБЛЕР и Си.
4. Назовите основные регистры управления АЦП. Объясните назначение, формат и способ занесения в регистры АЦП управляющих слов с применением языков АССЕМБЛЕР и Си.
5. Опишите основные регистры управления ЦАП и формат управляющих слов, которые используются при конфигурации ЦАП на языках АССЕМБЛЕР и Си.
6. Объясните процедуру фильтрации скользящим средним. Нарисуйте частотную и импульсную характеристику такого фильтра.

Список литературы

1. Семенов Б.Ю. Микроконтроллеры MSP430. Первое знакомство, М.: Изд-во «Солон-пресс», 2006, 120 с.
2. Шкелёв Е.И. Электронные цифровые системы и микропроцессоры. Учебное пособие. //Н.Новгород: Изд.ННГУ, 2004, 152 с.
3. Цифровые процессоры обработки сигналов. Справочник. // Под редакцией А.Г. Остапенко. – М.: Радио и связь, 1994.
4. Электронная версия руководства пользователя (оригинал) // http://www.ti.com/lit/ug/slau144j/slau144j.pdf

5. Семейство микроконтроллеров MSP430x2xx: Архитектура. Программирование. Разработка приложений. М.:"Додека XXI". 2010.
6. Электронный учебный курс по MSP 430 // http://we.easyelectronics.ru/blog/msp430
7. MSP430-P2618 development board. Users manual
8. http://lib.uni-dubna.ru/search/files/elec_msp430x1xx/~elec_msp430x1xx.htm
9. http://www.ti.com/product/MSP430F2618/description

Содержание

Евгений Иванович **Шкелёв**
Владимир Владимирович **Пархачёв**
Дмитрий Николаевич **Ивлев**
Виталий Юрьевич **Семенов**

РАБОТА С ЦАП И АЦП МИКРОКОНТРОЛЛЕРА СЕРИИ MSP430

Практикум

Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Нижегородский государственный университет им. Н.И. Лобачевского».
603950, Нижний Новгород, пр. Гагарина, 23.