

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования**

Санкт-Петербургский государственный университет
аэрокосмического приборостроения

Кафедра № 11 Аэрокосмических измерительно-вычислительных комплексов

Составитель доц. к.т.н. Малаханов Р.Н.

Система автоматического контроля

Лабораторная работа
по дисциплине «Цифровые вычислительные устройства и
микропроцессоры»

Санкт-Петербург

2020

Список сокращений

АЦП – аналого-цифровой преобразователь

АЦПр – аналого-цифровое преобразование

ДОП – демонстрационно-отладочная плата

ЕИ – единичный индикатор

МК – микроконтроллер

ПО – программное обеспечение

РАЦП – результат аналого-цифрового преобразования

РЦВВ – режим циклического включения-выключения

СИД – светоизлучающий диод

Цель работы заключается в ознакомлении с функциональным устройством модуля АЦП микроконтроллера 1986BE93У производства АО «ПКК Миландр» и принципами его программного управления.

Описание модуля АЦП приведено в спецификации на МК серии 1986BE9х [1] разделе 23 «Контроллер MDR_ADC».

Для работы с модулем АЦП предназначены следующие функции.

1. RST_CLK_PCLKcmd(RST_CLK_PCLK_ADC, ENABLE) – разрешение тактирования модуля АЦП.

2. ADC1_SetChannel(ADC_CH_ADC2) – установка канала модуля АЦП.

На ДОП потенциометер подключён ко второму входу АЦП.

3. ADC1_Cmd(ENABLE) – включение модуля АЦП.

4. ADC1_Start() – запуск АЦПр.

5. ADC1_GetFlagStatus(ADCx_IT_OUT_OF_RANGE) – определение значения разряда FlgREGAWOIFEN регистра MDR_ADC->ADC1_STATUS. Функция возвращает одну из двух констант: SET или RESET. Значение SET указывает на то, что разряд установлен: РАЦП вышел за установленные границы диапазона. Значение RESET указывает на то, что РАЦП находится внутри установленного диапазона.

6. ADC1_GetResult() – чтение РАЦП. Функция возвращает значение регистра MDR_ADC->ADC1_RESULT.

7. NVIC_EnableIRQ(ADC_IRQn) – разрешение прерываний от модуля АЦП. При возникновении прерывания будет происходить вызов обработчика прерывания - функции void ADC_IRQHandler(void).

8. NVIC_DisableIRQ(ADC_IRQn) – запрещение прерываний от модуля АЦП.

9. ADC1_ITConfig(ADCx_IT_END_OF_CONVERSION, ENABLE) – разрешение прерываний после окончания АЦПр.

10. ADC1_ITConfig(ADCx_IT_OUT_OF_RANGE, ENABLE) – разрешение прерываний при выходе напряжения из указанного диапазона.

11. ADC1_ITConfig(ADCx_IT_END_OF_CONVERSION | ADCx_IT_OUT_OF_RANGE, ENABLE) - разрешение прерываний после окончания АЦПр и при выходе напряжения из указанного диапазона.

12. void ADC_IRQHandler(void) – функция обработки прерываний модуля АЦП.

13. ADC1_LevelsConfig(LOW_LEVEL, HIGH_LEVEL, ADC_LEVEL_CONTROL_Enable) – установка значений напряжения для диапазона, при выходе из которого происходит запрос на прерывание. LOW_LEVEL устанавливает нижнее значение диапазона, а HIGH_LEVEL – верхнее.

14. ADC1_SamplingModeConfig(ADC_SAMPLING_MODE_CICLIC_CONV) – установка режима непрерывного АЦПр: после окончания одного преобразования сразу же запускается следующее.

15. ADC1_ClearOutOfRangeFlag() сбрасывает разряд FlgREGAWOIFEN в регистре ADC1_STATUS

Расчёт выходных значений АЦП

Цена единица младшего 12-разрядного АЦП определяется выражением:

$$\delta = (U_{\max} - U_{\min})/2^{12} - 1 = 3,3/4095 = 8,0586 \cdot 10^{-4} \text{ В},$$

где $U_{\max} = 3,3 \text{ В}$ и $U_{\min} = 0 \text{ В}$ – опорные напряжения АЦП. Выходное значение АЦП

$$N = \text{INT}(U_{\text{вх}} / \delta),$$

где $U_{\text{вх}}$ – значение входного напряжения на входе АЦП; функция $\text{INT}()$ возвращает целочисленное значение. Например, если $U_{\text{вх}} = 2,1 \text{ В}$, то этому напряжению соответствует целое значение

$$N = \text{INT}(2,1 / 8,0586 \cdot 10^{-4}) = 2606 = \text{A2Eh}.$$

Порядок выполнения контрольной работы

1. В соответствии со своим вариантом разработайте алгоритм работы программы для МК. Варианты индивидуальных заданий приведены в табл. 1.

Для студентов заочной формы обучения номер варианта совпадает с последней цифрой в зачётной книжке; если последней цифрой является нуль, то следует выполнять 10 вариант.

2. В соответствии с разработанным алгоритмом разработайте программу на языке программирования Си.

3. Произведите компиляцию программы, запрограммируйте МК и убедитесь в правильном выполнении программы, после чего покажите работу программы преподавателю.

4. Составьте отчёт о выполнении лабораторной работы.

ПО следует разрабатывать для ДОП 1986EvBrd_48, описание которой приведено в [2], а её электрическая принципиальная схема - в [3].

На ДОП два СИД, выполняющих функции ЕИ и обозначенных как *VD2* и *VD3* [3], подключены к выводам *PF0* и *PF1* порта *F* микроконтроллера соответственно (сигнальные цепи *PF0_VD* и *PF1_VD*). Для программного управления СИД выводы *PF0* и *PF1* должны быть настроены на работу в режиме цифрового вывода. После этого для включения ЕИ надо в первый или нулевой разряды регистра *RXTX* записать единицу, а для выключения – нуль.

Переменный резистор *R1* подключён ко второму входу АЦП (вывод *PD2* микроконтроллера). На ДОП переменный резистор *R1* имеет форму призмы синего цвета с винтом сверху. С помощью переменного резистора *R1* производится изменение значения напряжения на входе АЦП. При выходе значения за нижнюю границу СИД *VD3* должен переходить в РЦВВ (мигать). При выходе значения за верхнюю границу СИД *VD2* должен переходить в РЦВВ. При нахождении значения напряжения в указанном диапазоне оба СИД должны быть выключены. Значения границ диапазона приведены в таблице вариантов. РЦВВ производится с частотой, указанной в таблице вариантов.

В листинге 1 приведена программа управления первым АЦП и СИД для контроля уровня напряжения без программной реализации РЦВВ. Диапазон контролируемых значений от 1,1 до 2,1 В. Студенту необходимо самостоятельно разработать программное обеспечение для реализации РЦВВ с использованием системного таймера *SysTick*.

При создании проекта включите в него драйвер модуля АЦП (см. рис. 1). Для этого в разделе *Drivers* установите галочку в строке *ADC*.

Листинг программы управления АЦП

Листинг 1 – Программа управления модулем АЦП с генерацией прерываний при выходе значения напряжения из указанного диапазона.

Переменная *OutOfRange* устанавливается равной единице в ФОП при выходе значения напряжения за установленные границы.

```
#include "MDR32F9Qx_port.h"
```

```

#include "MDR32F9Qx_rst_clk.h"
#include "MDR32F9Qx_adc.h"

#define HIGH_LEVEL 0x0A2E // 2,1 B
#define LOW_LEVEL 0x0555 // 1,1 B

// Functions
void PortInit(void);

unsigned char OutOfRange;

int main(void)
{
    //static ADCx_InitTypeDef sADCx;

    PortInit();

    // ADC
    RST_CLK_PCLKcmd(RST_CLK_PCLK_RST_CLK | RST_CLK_PCLK_ADC,
ENABLE);
    NVIC_EnableIRQ(ADC_IRQn);
    //ADC1_ITConfig(ADCx_IT_END_OF_CONVERSION |
ADCx_IT_OUT_OF_RANGE, ENABLE);
    ADC1_ITConfig(ADCx_IT_OUT_OF_RANGE, ENABLE);
    ADC1_SetChannel(ADC_CH_ADC2);
    ADC1_LevelsConfig(LOW_LEVEL, HIGH_LEVEL,
ADC_LEVEL_CONTROL_Enable);
    ADC1_SamplingModeConfig(ADC_SAMPLING_MODE_CICLIC_CONV);
    ADC1_Cmd(ENABLE);

    OutOfRange = 0;
    while(1)
    {
        if(OutOfRange)
        {
            static unsigned a;

            OutOfRange = 0;
            a = ADC1_GetResult() & 0x0FFF;
            if(a > HIGH_LEVEL)
                PORT_SetBits(MDR_PORTF, PORT_Pin_0);
            if(a < LOW_LEVEL)
                PORT_SetBits(MDR_PORTF, PORT_Pin_1);
        }
        else
        {
            PORT_ResetBits(MDR_PORTF, PORT_Pin_0);
            PORT_ResetBits(MDR_PORTF, PORT_Pin_1);
        }
    }
}
// =====

```

```

void PortInit(void)
{
    static PORT_InitTypeDef PortInit;

    /* Enables clock on PORT F */
    RST_CLK_PCLKcmd(RST_CLK_PCLK_PORTF, ENABLE);

    /* Configure PORTF pins for output to switch LEDs on/off */
    PortInit.PORT_Pin = (PORT_Pin_0 | PORT_Pin_1);
    PortInit.PORT_OE = PORT_OE_OUT;
    PortInit.PORT_FUNC = PORT_FUNC_PORT;
    PortInit.PORT_MODE = PORT_MODE_DIGITAL;
    PortInit.PORT_SPEED = PORT_SPEED_SLOW;
    PORT_Init(MDR_PORTF, &PortInit);
}
// =====
void ADC_IRQHandler(void)
{
    if(ADC1_GetFlagStatus(ADCx_IT_OUT_OF_RANGE))
    {
        OutOfRange = 1;
        // Сброс разряда FlgREGAWOIFEN в регистре ADC1_STATUS
        ADC1_ClearOutOfRangeFlag();
    }
}

```

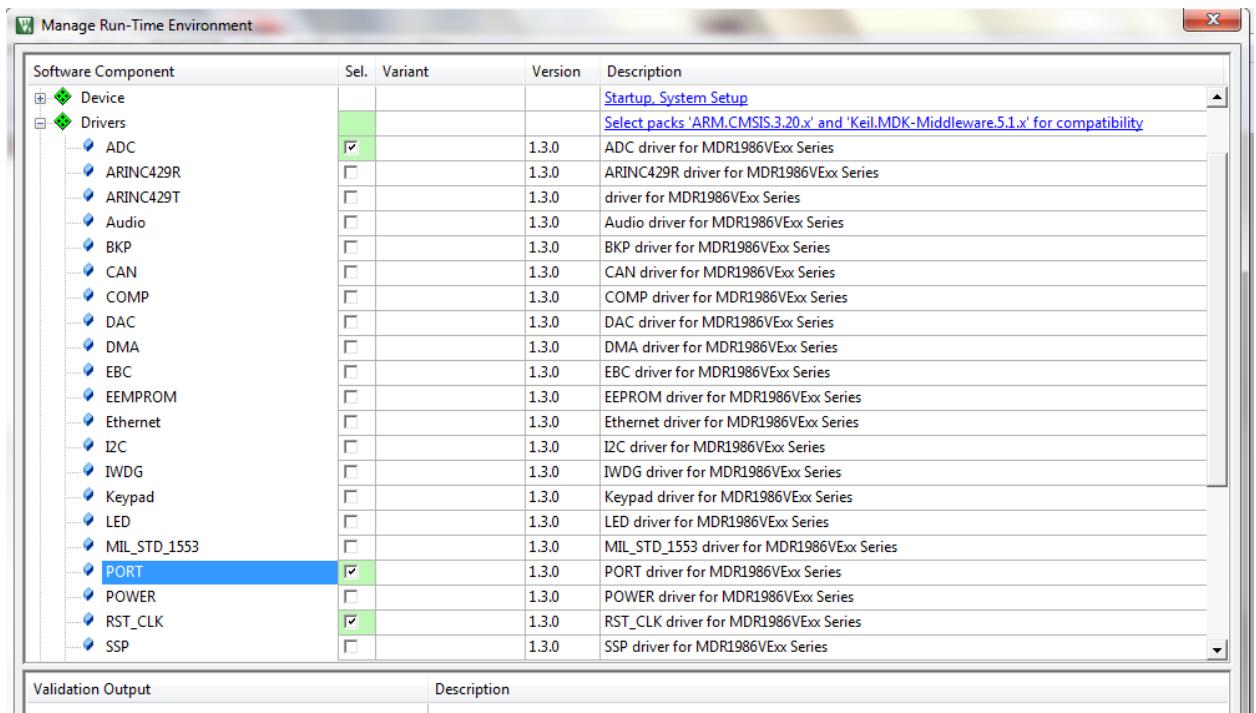


Рисунок 1 – Окно выбора компонентов для проекта.

Таблица 1.

Вариант	Значения диапазона, В		Частота индикации, Гц	Примечание
	нижнее	верхнее		
1	0,5	2,8	0,5	
2	2	3,1	1	
3	0,7	2,6	2	
4	2,5	3	0,5	
5	0,9	2,5	1	
6	1	2,4	2	
7	1,2	2,2	0,5	
8	1,3	2	1	
9	1,5	2	2	
10	1,7	2,7	0,5	

Оформление отчёта

Отчёт должен быть оформлен в соответствии с требованиями нормоконтроля и должен содержать:

- цель работы;
- задание;
- блок-схему алгоритма программы;
- текст программы на языке программирования Си;
- библиографический список;
- выводы о проделанной работе.

Библиографический список

1. Микросхема 32-разрядного однокристального микро-ЭВМ с памятью Flash-типа 1986VE9ху, К1986VE9ху, К1986VE92QI, К1986VE92QC, 1986VE91Н4, К1986VE91Н4, 1986VE94Н4, К1986VE94Н4 [Электронный ресурс]: Спецификация. – Версия 3.20.1 от 15.05.2020. – Электрон. дан. (6,89 Mbytes). – [Б.м.]: АО «ПКК Миландр», 2020. – Режим доступа: <http://ic.milandr.ru/upload/iblock/33e/33e6826dd444292426050ff6e39095a1.pdf>, свободный. – Загл. с экрана.

2. Демонстрационно-отладочная плата 1986EvBrd_48 [Электронный ресурс] : Техническое описание. – Версия 1.0 от 25.05.2010. – Электрон. дан. (1,05 Mbytes). – [Б.м.]: ЗАО «ПКК Миландр», 2010. – 1 электрон. опт. диск (CD-ROM). – Систем. требования: Pentium 100МГц ; 16 Мб RAM ; Windows 7 ; CD-ROM дисковод ; SVGA видеокарта, 256 цв. ; мышь. - Загл. с экрана. - CD-ROM входит в комплект поставки демонстрационно-отладочной платы 1986EvBrd_48.

3. Отладочная плата 1986BE93У [Электронный ресурс] : Схема электрическая принципиальная. – Revision 3. Последнее изменение 24.04.2014. – Электрон. граф. дан. (148 Kbytes). – [Б.м.]: ЗАО «ПКК Миландр», 2014. - 1 электрон. опт. диск (CD-ROM). – Систем. требования: Pentium 100МГц ; 16 Мб RAM ; Windows 7 ; CD-ROM дисковод ; SVGA видеокарта, 256 цв. ; мышь. – Загл. с экрана. - CD-ROM входит в комплект поставки демонстрационно-отладочной платы 1986EvBrd_48.