

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Политехнический институт
Факультет «Энергетический»
Кафедра «Автоматизированный электропривод»

ОТЧЕТ
по лабораторной работе №5

АЦП

Проверил:

_____ /Качалов А.В./
«__» _____ 2020 г.

Автор работы
студент группы П-

_____/./
«__» _____ 2020 г.

Челябинск 2020

Цель работы

Освоить теоретический и практический материал по работе 10-разрядного аналого-цифрового преобразователя (АЦП) микроконтроллера AtmegaXX. Применить приобретенные навыки при написании программы.

Созыкин Дмитрий Дмитриевич
18.05.99

Данные согласно варианту
3 канал АЦП

Левое выравнивание

Кд = 128

4 выходных разряда

Режим управления – несимметричный

ШИМ – быстрый, 8 бит

Номинальная скорость = 1700

Диапазон изменения скорости -1350 0 1350

Функциональная схема работы устройства

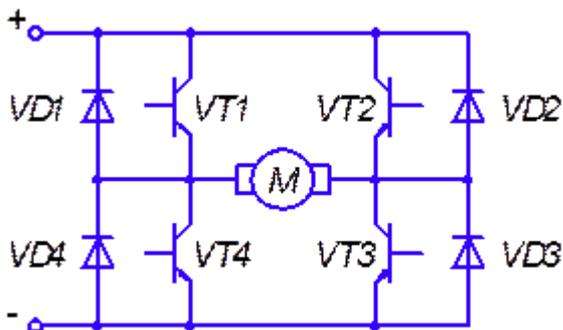


Рисунок 1 – Функциональная схема

Расчёт регистров ШИМ

Расчёт начинается с выбора режима в котором будет работать таймер. Режим ШИМ, фазовый.

Максимальная скорость = 1700 Об\мин. Соответственно для получения максимальной скорости, заданной по варианту необходимо произвести следующее вычисление.

$\text{max speed} = 1700 \setminus 1350 = 0.8$. Данная скорость составляет 0.8 от максимально заполненного ШИМ. Т.е т.к ШИМ 8 бит, то максимальная заполненность достигает, когда в регистре OCR находится число 255. Соответственно по варианту 0.8 от 255+1 составляет 205 от maxPWM (256)

; всего 16 скоростей т.к. значение 4ех разрядное согласно варианту, следовательно $205 \setminus 16 = 13$ от maxspeed на каждую скорость 13
Соответственно для получения скорости необходимо умножить полученную скорость на константу 13.

Настройка регистров АЦП

Согласно варианту присутствует левое выравнивание, соответственно бит ADLAR в регистре ADMUX должен быть установлен, а полученный результат сдвинут на 4 бита. Также для выбора первого канала регулирования необходимо установить бит 0 и первый бит в битах MUX. Также для выбора предделителя установим биты ADPS2,1,0 в регистре ADCSRA, установим 6 и 7 бит этого регистра, чем разрешим работу АЦП и запустим преобразование

Листинг программы

;Программа для управления электродвигателем постоянного тока.

```
.equ EN_WORK = 0
.equ REVERS = 1
.equ BIT_MASK = $03
.equ SPEED_MASK = $06
.equ CONST = 13

.def SPEED_REG = R18

.include "m32def.inc" ;подключение стандартной библиотеки
.cseg ;начало сегмента кода
.org $0
reset:
    ldi r16,low(RAMEND) ;Инициализация стека
    ldi r17,high(RAMEND)
    out spl,r16
    out sph,r17

    clr r16;Инициализация портов ввода/вывода
    out DDRA,r16 ;Порт A - на ввод информации
    ser r16
    out DDRD,r16 ;Порт D - на вывод информации.
    out DDRB,r16 ;Порт B - на вывод информации.

    clr r16
    out OCR1AH,r16 ;обнуление регистров T1
    out OCR1BH,r16
    out TCNT1H,r16
```

```
out TCNT1L,r16
ldi r16,0xA1 ; инициализация таймера ШИМ 8 бит, быстрый
out TCCR1A,r16
ldi r16,0x0B ;
out TCCR1B,r16
```

```
clr SPEED_REG
rcall startADC
```

main:

```
in r16,ADCSRA ;Опрос регистра ADCSRA
SBRC r16,4 ;Если бит 4 чист, то следующая команда пропускается
rcall ADC_ready ;Иначе вызов ADC_ready
rjmp main ;Возврат на main
```

ADC_ready:

```
in r23,SREG ;сохранение значения регистра SREG
```

```
ldi r16,0x97 ;Обнуление флага ADIF записью в него логической«1»
out ADCSRA,r16
```

```
in r16, PINA ; маска на вход
andi r16, BIT_MASK
sbrs r16, EN_WORK ; проверка разрешения
rjmp stop ; нет разрешения останов двигателя
rcall set_speed ; установка скорости:
out PORTB, SPEED_REG ; вывод скорости
sbrc r16, REVERS
rjmp revers_set ; если есть реверс, то переход на установки реверса
clr r17
out OCR1BL, R17 ; если реверса нет, то идем дальше
out OCR1BH, R17
out OCR1AL, SPEED_REG
```

quit:

```
ldi r16,0xA1 ; ; установка значений таймера и АЦП
out TCCR1A,r16
ldi r16,0x0B ;
out TCCR1B,r16
```

```
rcall startADC
out SREG, r23
```

```
rjmp main
```

```
revers_set: ; установки таймера для реверса
```

```
clr r17
```

```
out OCR1AL, R17
```

```
out OCR1AH, R17
```

```
out OCR1BL, SPEED_REG
```

```
rjmp quit
```

```
stop: ; Останов двигателя. Либо нет сигнала вкл. Либо не нажата скорость
```

```
clr r16
```

```
out TCCR1A, r16
```

```
out TCCR1B, r16
```

```
out PORTB, r16 ; скорость очистили
```

```
out ADCSRA, r16 ; выключили бит 7. ADEN
```

```
rjmp main
```

```
set_speed:
```

```
in r17, ADCH ; Считывание результата преобразования из ADCH
```

```
lsr r17 ; 4 разряда
```

```
lsr r17
```

```
lsr r17
```

```
lsr r17 ; max speed =  $1350 \setminus 1700 = 0.8 = 205$  от maxPWM (256)
```

```
; всего 16 скоростей  $205 \setminus 16 = 13$  от max speed на каждую скорость 13
```

```
ldi r21, CONST
```

```
mul r17, r21
```

```
mov SPEED_REG, r0
```

```
ret
```

```
startADC:
```

```
ldi r16, 0x63 ; Инициализация мультиплексора АЦП
```

```
out ADMUX, r16
```

```
ldi r16, 0xC7
```

```
out ADCSRA, r16 ; Инициализация АЦП и его запуск
```

```
ret
```

Вывод: Был исследован аналого-цифровой преобразователь.