**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

**УПРАВЛЕНИЕ ПЕРФИРЕЙНЫМИ УСТРОЙСТВАМИ ПРИ ПОМОЩИ ИНКРЕМЕНТАЛЬНОГО ЭНКОДЕРА**

**Цель работы**

Ознакомиться с функциональными возможностями и внутренней структурой учебно-отладочного стенда EV8031/AVR. Изучить внутреннюю организацию и принципы работы инкрементального энкодера. Научиться управлять различными периферийными устройствами стенда при помощи инкрементального энкодера.

**1 Краткие теоретические сведения**

Энкодер (преобразователь угловых перемещений) – это устройство, предназначенное для преобразования угла поворота вращающегося объекта (вала) в электрические сигналы, позволяющие определить угол его поворота.

Энкодеры очень широко применяются в промышленности. Они подразделяются на инкрементальные и абсолютные, которые могут достигать очень высокого разрешения.

Инкрементальный энкодер выдает за один оборот определенное количество импульсов, абсолютные энкодеры позволяют в любой момент времени знать текущий угол поворота оси, в том числе и после пропадания и восстановления питания. Многооборотные абсолютные энкодеры, кроме того, также подсчитывают и запоминают количество полных оборотов оси.

Энкодеры могут быть как оптические, резистивные, так и магнитные, при этом они могут работать через шинные интерфейсы или промышленную сеть.

Преобразователи угловых перемещений на сегодняшний день практически полностью вытеснили применение сельсинов.

**1.1 Инкрементальные энкодеры**

Инкрементальные энкодеры предназначены для определения угла поворота вращающихся объектов. Они генерируют последовательный импульсный цифровой код, содержащий информацию относительно угла поворота объекта. Если вал останавливается, то останавливается и передача импульсов. Основным рабочим параметром инкрементального энкодера является количество импульсов за один оборот. Мгновенную величину угла поворота объекта определяют посредством подсчёта импульсов от старта. Для вычисления угловой скорости объекта вычислительный элемент тахометра выполняет дифференцирование количества импульсов во времени, таким образом, показывая сразу величину скорости, то есть число оборотов в минуту.

Выходной сигнал инкрементального энкодера имеет два канала, в которых идентичные последовательности импульсов сдвинуты на 90° относительно друг друга (так называемые парафазные импульсы), что позволяет определять направление вращения. Имеется также цифровой выход нулевой метки, который позволяет всегда рассчитать абсолютное положение вала.

Импульсный (или инкрементальный, пошаговый) энкодер относится к типу энкодеров, которые предназначены для указания направления движения и/или углового перемещения внешнего механизма. Инкрементальный энкодер формирует импульсы, количество которых соответствует повороту вала на определенный угол. Этот тип энкодеров, в отличие от абсолютных, не формирует код положения вала, когда вал находится в покое. Энкодер связан со счетным устройством, это необходимо для подсчета импульсов и преобразования их в меру перемещения вала.

Пошаговый оптический энкодер состоит из следующих компонентов: источника света, диска с метками, фототранзисторной сборки и схемы обработки сигнала. Диск пошагового энкодера (смотри рисунок 1.1) подразделен на точно позиционированные отметки. Количество отметок определяет количество импульсов за один оборот. К примеру, если диск поделен на 1000 меток, тогда за 250 импульсов вал должен повернуться на 90 градусов.

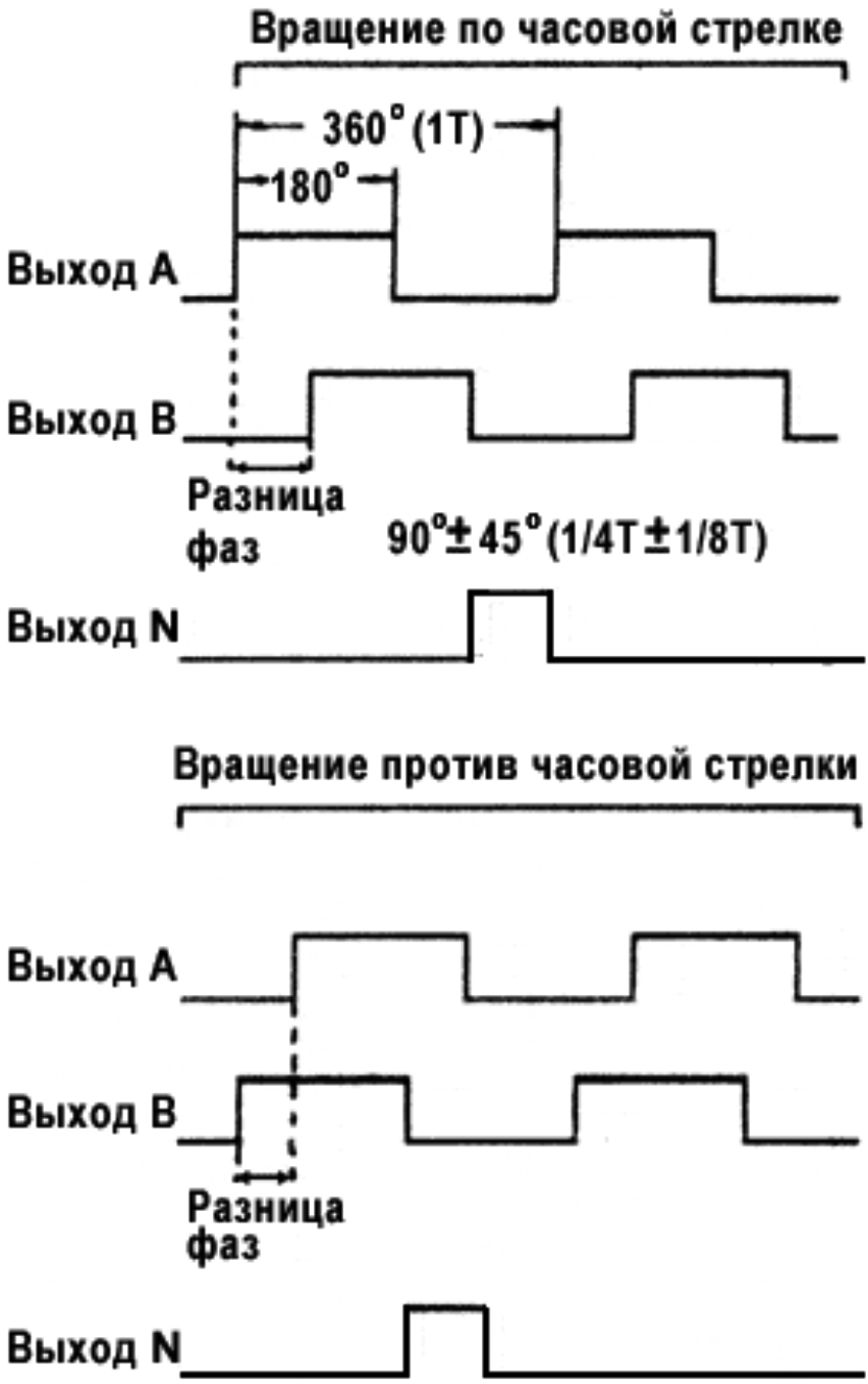


Рисунок 1 – Сдвиг фаз между каналами А и В при разных направлениях вращения инкрементального энкодера

Однооборотными (или single-turn) энкодерами называются такие датчики, которые выдают абсолютное значения в пределах одного полного оборота (т.е. на все 360°). После одного оборота код является полностью пройденым и начинается опять с его начального значения. Эти датчики служат, преимущественно, для измерения угла поворота и применяются, например, в антенных системах, эксцентричных коленчатых прессах и т.д. Внутренне устройство однооборотного энкодера изображено ниже, на рисунке 1.5.

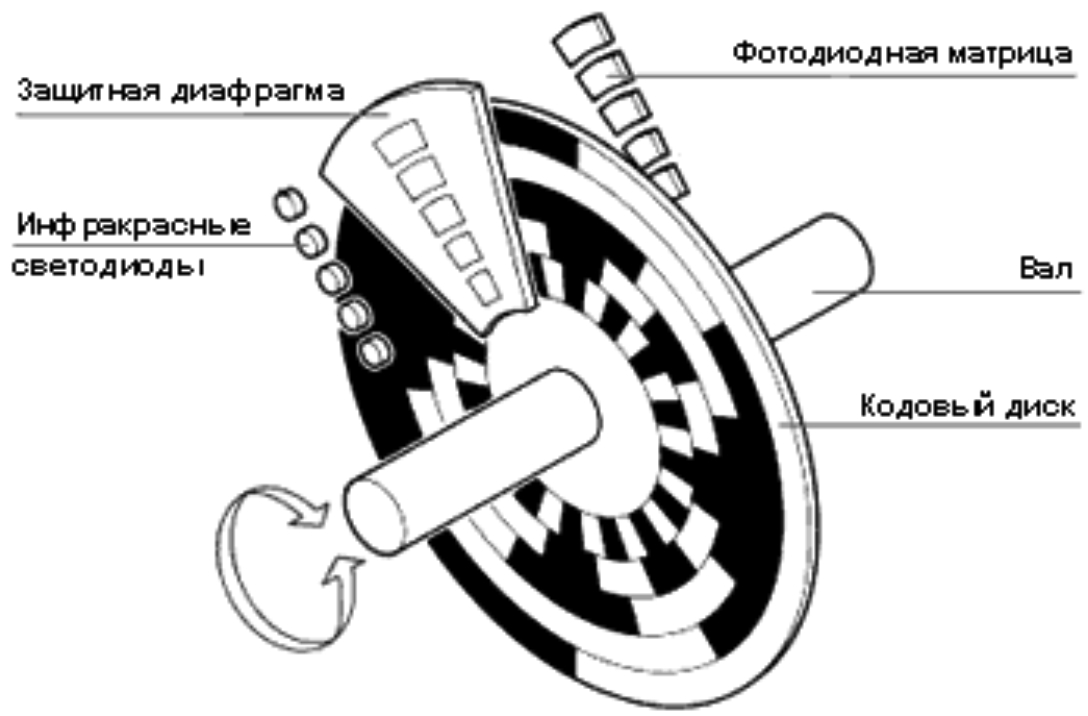


Рисунок 2 – Внутренне устройство однооборотного энкодера

Линейные перемещения предполагают необходимым применение измерительной системы с n-количеством оборотов. Например, при линейных приводах или при задачах измерения с помощью зубчатой измерительной штанги, применение однооборотных датчиков является неприемлемым. В этом случае приходят на помощь датчики, где дополнительно к измерению угла поворота в пределах одного оборота также происходит регистрация количества оборотов с помощью дополнительно встроенного передаточного механизма, т.е. своего рода редуктора из нескольких кодовых оптических дисков, образуя, таким образом, многооборотный энкодер (или multi-turn). Внутренне устройство многооборотного энкодера изображено ниже, на рисунке 1.6.

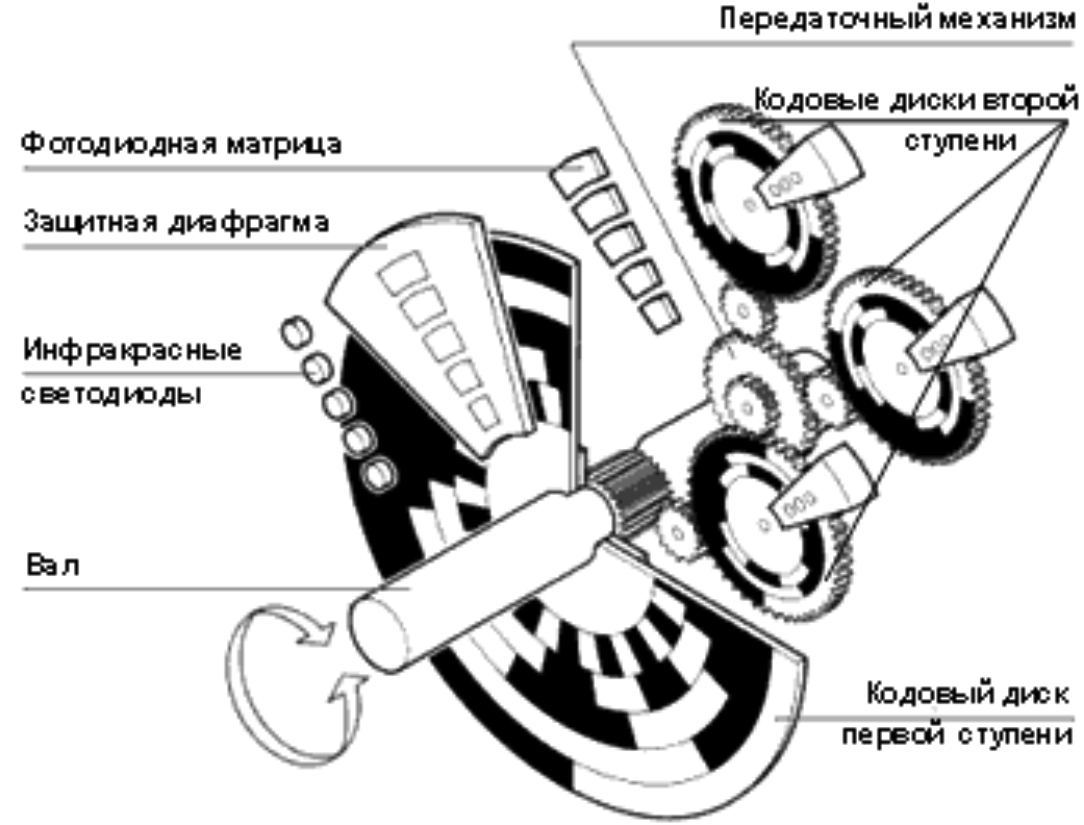


Рисунок 3 – Внутреннее устройство многооборотного энкодера

Оптические энкодеры имеют жёстко закреплённый соосно валу стеклянный диск с прецизионной оптической шкалой. При вращении объекта оптопара считывает информацию, а электронная схема преобразовывает её в последовательность дискретных электрических импульсов. Абсолютные оптические энкодеры – это датчики угла поворота, где каждому положению вала соответствует уникальный цифровой выходной код, который наряду с числом оборотов является основным рабочим параметром датчика. Абсолютные оптические энкодеры, так же как и инкрементальные энкодеры, считывают и фиксируют параметры вращения оптического диска. Внутреннее устройство оптического энкодера изображено ниже, на рисунке 1.7.

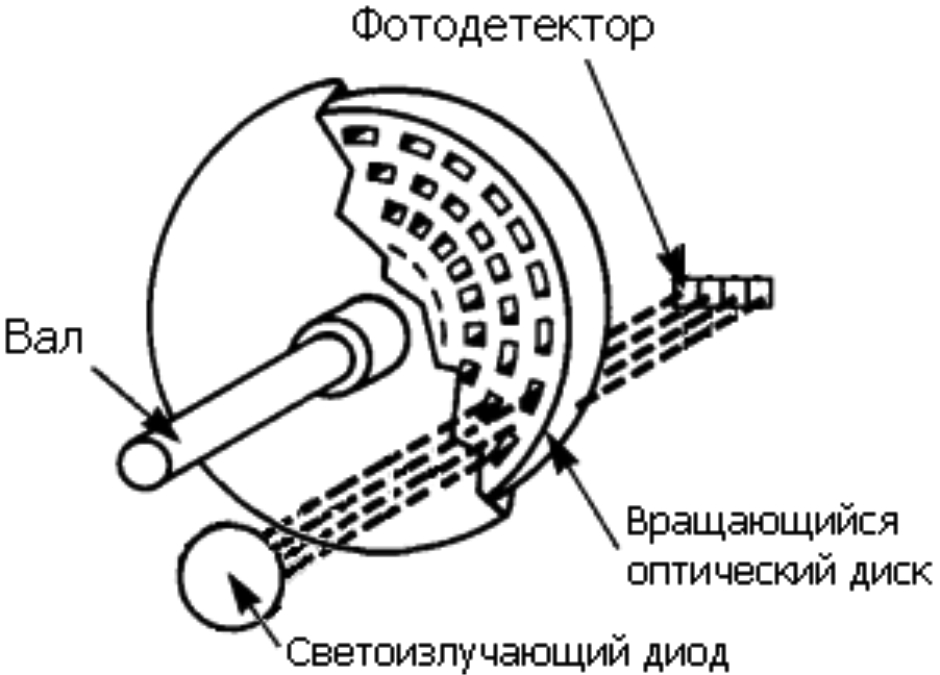


Рисунок 4 – Внутреннее устройство оптического энкодера

Магнитные энкодеры с высокой точностью регистрируют прохождение магнитных полюсов вращающегося магнитного элемента непосредственно вблизи чувствительного элемента (посредством датчика Холла), преобразуя эти данные в соответствующий цифровой код. Внутреннее устройство магнитного энкодера изображено ниже, на рисунке 1.8.

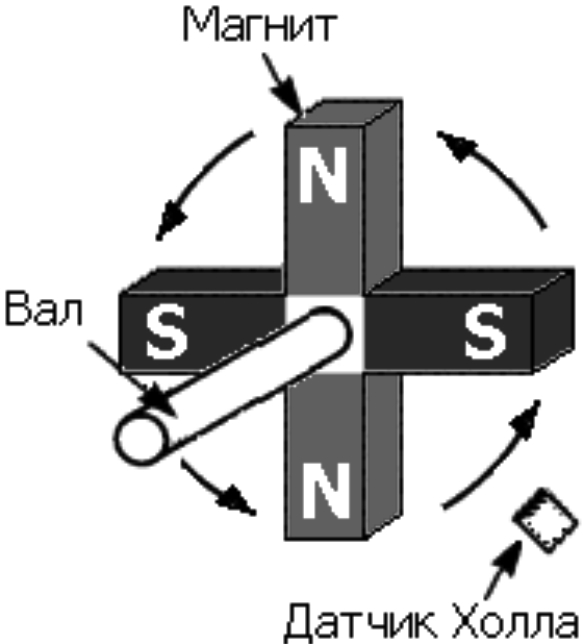


Рисунок 5 – Внутреннее устройство магнитного энкодера

Магниторезистивный энкодер состоит из катушки, помещенной в магнитное поле, катушка закрепляется на валу. При вращении катушки ее витки будут изменять положение относительно поля, они будут то параллельны полю, то перпендикулярны ему, соответственно ток в катушке будет изменяться. Таким образом, протекающий через катушку ток будут изменяться в зависимости от угла поворота вала. Внутреннее устройство магниторезистивного энкодера изображено ниже, на рисунке 1.9.

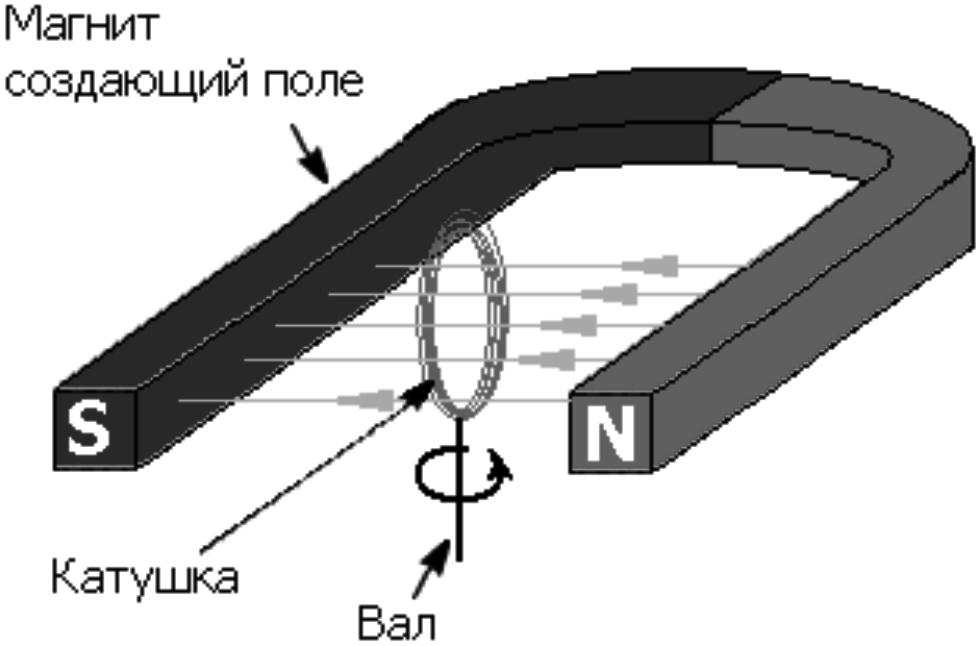


Рисунок 6 – Внутреннее устройство магниторезистивного энкодера

Механические и оптические энкодеры с последовательным выходом содержат диск из диэлектрика или стекла с нанесёнными выпуклыми, проводящими или непрозрачными участками. Считывание абсолютного угла поворота диска производится линейкой переключателей или контактов в случае механической схемы и линейкой оптронов в случае оптической. Выходные сигналы представляют собой код Грея, позволяющий избавиться от неоднозначности интерпретации сигнала.

Соединение энкодера с вращающимся объектом осуществляется посредством нормального или полого вала, последний может быть как сквозным, так и несквозным (тупиковым). Вал вращающегося объекта и вал энкодера соединяют механически при помощи гибкой или жёсткой соединительной муфты. В качестве альтернативы энкодер монтируют непосредственно на вал объекта, если энкодер имеет полый вал. В первом случае вероятная несоосность и допустимые биения компенсируются деформацией гибкой втулки. Во втором возможна фиксация энкодера посредством штифта.

**2 Порядок выполнения работы**

2.1 Запустить IDE AVR Studio 4.

2.2 Создать новые проекты в IDE AVR Studio 4.

2.3 В появившихся окнах написать программы на языках С и ассемблер с учётом варианта задания, который указан в таблице 2.1.

Таблица 1 – Варианты заданий

|  |  |
| --- | --- |
| № | Описание задания |
| 2 | При каждом повороте энкодера по часовой стрелке на 6 щелчков значение напряжения, подаваемое на лампы, увеличивается на 0,5 В, а при повороте энкодера против часовой стрелки на 6 щелчков – уменьшается на 0,5 В. При нажатии на инкрементальный энкодер как на кнопку, лампы светятся с максимальной яркостью. Временные задержки организовываются программно. |

2.4 Произвести компиляцию проектов.

2.5 При наличие сообщений об ошибках или предупреждениях вернуться к предыдущему пункту и внести необходимые исправления. В случае некорректной работы программы выполнить её отладку средствами меню Debug.

2.6 Проверить подключение USB-кабеля программатора к одноимённому разъёму системного блока.

2.7 Загрузить исполняемые файлы проектов в микроконтроллер.

2.8 Визуально оценить правильность работы написанной программы.

**3 Исходные тексты программ на языках С и ассемблер**

#define F\_CPU 7372800L //?????? ??????? ?????? (7,3728 ???)

#include <avr/io.h>

#include <avr/iom8515.h>

#include <util/delay.h>

#define LAMP 0xF000 //????? ??????? ??????????? ? ??????

//+25

int main(void) {

ACSR= 1<<ACD; //?????????? ??????? ??????????? ???????????

MCUCR= 1<<SRE; //????????? ?????? ? ??????? ???????

volatile unsigned char \*lamp;

lamp= (unsigned char\*) LAMP;

unsigned char enc\_pressed = 0;

unsigned char dac\_value = 0;

unsigned char rot\_dec = 0;

unsigned char rot\_inc = 0;

unsigned char next\_encoder\_state;

unsigned char last\_encoder\_state= PINB;

last\_encoder\_state &= 0b1001000;

while(1) {

\_delay\_ms(1);

next\_encoder\_state = PINB;

next\_encoder\_state &= 0b10010000;

if(next\_encoder\_state!= last\_encoder\_state) {

if(last\_encoder\_state== 0x00) {

if(next\_encoder\_state == 0b00010000) {

++rot\_dec;

if (rot\_dec == 5){

dac\_value += 25;

rot\_dec = 0;

}

}

if(next\_encoder\_state == 0b10000000) {

++rot\_inc;

if (rot\_inc == 5){

dac\_value -= 25;

rot\_inc = 0;

}

}

}

}

\*lamp = dac\_value;

enc\_pressed = PINB;

enc\_pressed &= 0b00100000;

if(enc\_pressed == 0x00) {

dac\_value = 0xFF; //??????????, ????? ????????? ????????

\*lamp = dac\_value;

}

last\_encoder\_state = next\_encoder\_state;

}

return 0;

.include "m8515def.inc";??????????? ?????? ??????????? ATmega8515

.def dac\_value = r16 ;???????, ???????? ????????? ???????????? ???????

.def temp = r17 ;??????? ?????????? ????????

.def rot\_dec = r18 ;??????? ????????? ????????

.def last\_encoder\_state = r19

.def next\_encoder\_state = r20

.def rot\_inc = r21

.def counter = r23 ;??????? ????? ? ???????????? ????????? ????????

.def long\_delay\_low =r24 ;??????? ???? ???????? ??????? ????????

.def long\_delay\_high =r25 ;??????? ???? ???????? ??????? ????????

.EQU DAC\_ADRESS = 0xF000

.CSEG

.ORG 0x0000

rjmp Init ; ?????? ?????????? ?? ??????

reti;rjmp EXT\_INT0 ; IRQ0 Handler

reti;rjmp EXT\_INT1 ; IRQ1 Handler

reti;rjmp TIM1\_CAPT ; Timer1 Capture Handler

reti;rjmp TIM1\_COMPA ; Timer1 Compare A Handler

reti;rjmp TIM1\_COMPB ; Timer1 Compare B Handler

reti;rjmp TIM1\_OVF ; Timer1 Overflow Handler

reti;rjmp TIM0\_OVF ; Timer0 Overflow Handler

reti;rjmp SPI\_STC ; SPI Transfer Complete Handler

reti;rjmp USART\_RXC ; USART RX Complete Handler

reti;rjmp USART\_UDRE ; UDR0 Empty Handler

reti;rjmp USART\_TXC ; USART TX Complete Handler

reti;rjmp ANA\_COMP ; Analog Comparator Handler

reti;rjmp EXT\_INT2 ; IRQ2 Handler

reti;rjmp TIM0\_COMP ; Timer0 Compare Handler

reti;rjmp EE\_RDY ; EEPROM Ready Handler

reti;rjmp SPM\_RDY ; Store Program memory Ready

Init:

ldi temp,low(RAMEND)

out SPL,temp

ldi temp,high(RAMEND)

out SPH,temp

sbi ACSR,7

ldi temp, 0b10000000 ;????????? ?????? ? ??????? ???????

out MCUCR, temp

ldi ZL, low(DAC\_ADRESS)

ldi ZH, high(DAC\_ADRESS)

clr dac\_value

clr rot\_dec

clr rot\_inc

in last\_encoder\_state, PINB

andi last\_encoder\_state, 0b10010000

Infinite\_loop: ;??????????? ????

rcall long\_delay

in next\_encoder\_state, PINB

andi next\_encoder\_state, 0b10010000

cp next\_encoder\_state, last\_encoder\_state

breq Next\_iterate

cpi last\_encoder\_state, 0b00000000

brne Next\_iterate ;????????? ? ?????????? ?????????

cpi next\_encoder\_state, 0b00010000

breq Decr\_state

cpi next\_encoder\_state, 0b10000000 ;???? ? ?????? A ???????? ????????? "1", ? ? ?????? B ??????? "0", ?? ??????? ????????? ?? ??????? ???????

breq Incr\_state ;????????? ? ?????????? ????????? ????????

rjmp Next\_iterate ;???? ?? ? ????? ?? ??????? ?? ????????? ????? - ???? ?????

Incr\_state:

inc rot\_dec

cpi rot\_dec, 5

brne Next\_iterate

ldi temp, 25

add dac\_value, temp

clr rot\_dec

rjmp Next\_iterate ; ????????? ? ?????????? ?????????

Decr\_state:

inc rot\_inc

cpi rot\_inc, 5

brne Next\_iterate

subi dac\_value, 25

clr rot\_inc

rjmp Next\_iterate

Next\_iterate:

in temp, PINB

andi temp, 0b00100000

tst temp

brne Led\_action ;???? ??? - ???? ??????

ldi dac\_value, 0xFF

Led\_action:

st Z, dac\_value

mov last\_encoder\_state, next\_encoder\_state

rjmp Infinite\_loop

long\_delay:

ldi long\_delay\_low,0x09

ldi long\_delay\_high,0x00

long\_loop:

rcall short\_delay ;???????? ????????

sbiw long\_delay\_high:long\_delay\_low,0b00000001

brne long\_loop ;???? ?? 0,????????? ????

ret;

short\_delay:

nop

ldi counter,0xC5 ;??????? ?????

short\_loop:

nop

dec counter

brne short\_loop

ret

.EXIT;????? ?????????

**9.4 Особенности IDE AVRStudio выявленные в ходе выполнения расчетно-графической работы**

В ходе расчетно-графической работы никаких новых особенностей IDE AVRStudio не было выявлено.

**Выводы**

В ходе лабораторной работы были изучены принципы работы с инкрементальным энкодером в учебно отладочном стенде EV8031/AVR. Была написана программа, которая производит управление яркостью лампы накаливания с помощью инкрементального энкодера. Также были рассмотрены иные типы энкодеров и принципы их конструкции.