МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ЧЕРНИГОВСКИЙ НАЦИОНАЛЬНЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ И КОМПЬЮТЕРНЫХ СИСТЕМ

**ОТЧЕТ**

По проектно-технологической практике.

Изучение методов цифро-аналогового преобразования

Исполнители:

студенты группы КИ-134 Кулик И. С.

Савосько С. С.

Принял:

Прелая О.А.

Чернигов 2016

**1 ЦИФРОАНАЛОГОВОЕ ПРЕОБРАЗОВАНИЕ. УПРАВЛЕНИЕ ПЕРИФЕРИЙНЫМИ УСТРОЙСТВАМИ ОТЛАДОЧНОГО СТЕНДА EV8031/AVR**

**1.1 Цель работы**

Изучить методы цифро-аналогового преобразования. Ознакомиться с функциональными возможностями и внутренней структурой отладочного стенда EV8031/AVR. Изучить внутреннюю структуру и организацию дополнительной платы расширения. Научиться управлять скоростью вращения электродвигателя постоянного тока (вентилятора) или яркостью свечения ламп по заданному алгоритму.

**1.2 Краткие теоретические сведения**

1.2.1 Цифро-аналоговое преобразование

Цифро-аналоговые преобразователи (ЦАП) вырабатывают напряжение или ток, функционально связанные с управляющим кодом. Применяются ЦАП для формирования выходных аналоговых сигналов цифровых измерительных и вычислительных устройств. Для преобразования двоичного кода в аналоговый сигнал обычно формируются токи, пропорциональные весам разрядов кода, и затем суммируются те из токов, которые соответствуют ненулевым разрядам входного кода.

Применяются в основном два метода ЦАП: суммирование единичных эталонных величин и суммирование эталонных величин, веса которых различаются. В первом при формировании выходной аналоговой величины используется только одна эталонная величина весом в один квант. Во втором методе применяются эталонные величины с весами, зависящими от номера разряда, и в суммировании участвуют только те эталонные величины, для которых в соответствующем разряде входного кода установлена единица.

При использовании на входе двоичного позиционного кода значения всех разрядов поступает одновременно, и работа таких ЦАП описывается выражением:

(1.1)



где X – аналоговая величина;

аi – коэффициенты соответствующих двоичных разрядов, которые принимают дискретные значения – единица или ноль;

Р – опорный сигнал;

b – число разрядов.

В преобразователях из опорного напряжения формируются эталонные величины, соответствующие значениям разрядов входного кода, которые суммируются и образуют дискретные значения выходной аналоговой величины.

Классификация ЦАП может быть проведена по следующим признакам:

а) способ формирования выходного напряжения – с суммирование напряжений, делением напряжений, суммированию токов;

б) вид выходного сигнала – с токовым выходом, с выходом по напряжению;

в) полярность выходного напряжения – постоянное, переменное.

Основные структуры, используемые в ЦАП интегрального исполнения, – это структуры с суммированием токов:

а) ЦАП – со взвешенными резисторами в цепях эмиттеров;

б) ЦАП – со взвешенными резисторами в цепях нагрузки;

в) ЦАП с лестничной матрицей R=2R в цепях эмиттеров транзисторов источников токов;

г) ЦАП с выходной лестничной матрицей R=2R.

К основным параметрам ЦАП относят:

а) n – число разрядов входного управляющего кода;

б) номинальный выходной ток;

в) время установления выходного сигнала после изменения входного управляющего кода;

г) погрешность полной шкалы;

д) погрешность линейности;

е) дифференциальная нелинейность.

Погрешности ЦАП могут быть выражены в процентах или других относительных единицах, а также в долях кванта.

Упрощенная схема ЦАП приведена на рисунке 1.1.

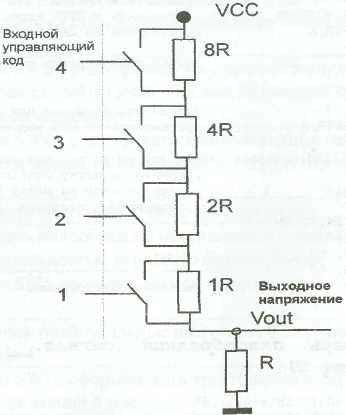


Рисунок 1.1 – Упрощённая схема работы цифро-аналогового преобразователя

Если соотношение резисторов ЦАП как 8R, 4R, 2R, 1R, то при включении всех коммутаторов, в соответствии с законом Ома напряжение в точке OUT будет равняться напряжению VCC. Аналогично можно вычислить значение напряжения, при различных комбинациях входного кода.

В составе учебно-отладочного стенда EV8031/AVR имеется микросхема ЦАП AD7801. Она представляет собой восьмиразрядный ЦАП (с параллельным интерфейсом) с размахом выходного напряжения равным напряжению питания. Доступ к ЦАП осуществляется как к ячейке внешнего ОЗУ по адресу 0F000h, причём на всех платах расширения.

1.2.2 Организация подключения и управление электрическим двигателем постоянного тока и лампами накаливания, входящими в состав учебно-отладочного стенда EV8031/AVR

В учебно-отладочном стенде EV8031/AVR (на одной из плат расширения) предусмотрена возможность работы микроконтроллера с электродвигателем постоянного тока и лампами накаливания.

Для управления яркостью свечения ламп накаливания либо скоростью вращения электродвигателя постоянного тока микроконтроллер формирует цифровой код, поступающий на вход ЦАП (доступ к ЦАП осуществляется как к ячейке внешнего ОЗУ по адресу 0F000h). На выходе ЦАП формируется аналоговый сигнал с уровнем напряжения, находящимся в диапазоне от +0В до Vcc (+5В). Находящийся на плате расширения ЦАП построен на микросхеме AD7801 и имеет разрядность входных данных 8 бит, т.е. минимальный цифровой код, загружаемый в него, равен 0, максимальный – 255. Для того, что вычислить, какой уровень выходного напряжения (Vadc) соответствует заданному на входе цифровому коду n, необходимо руководствоваться следующей формулой:

Vadc = Vcc\*(n/255) (1.2)

Выход ЦАП при помощи перемычки (джампера) JP3 может быть соединён со входом линейного регулятора DA1-1, построенного на операционном усилителе AD8532. Схема сопряжения ЦАП и линейного регулятора показана на рисунке 1.2.

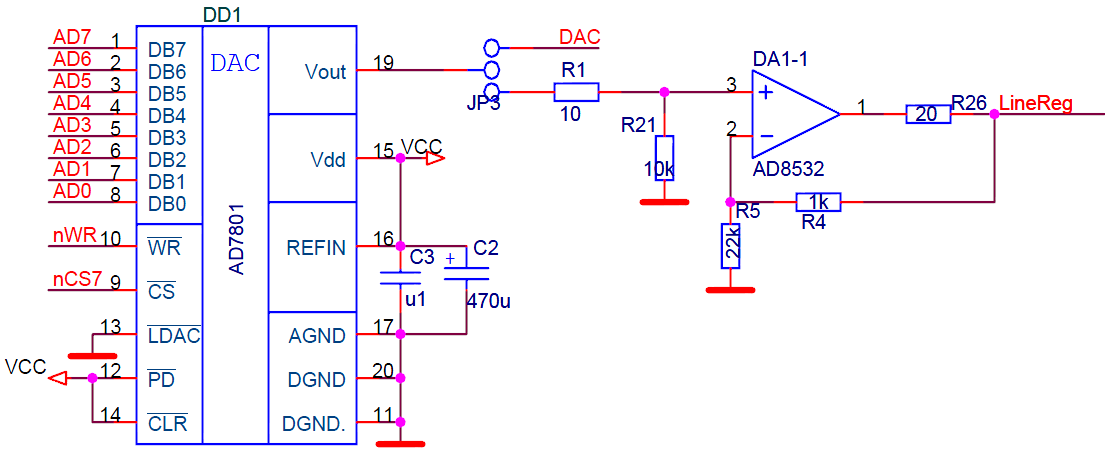


Рисунок 1.2 – Сопряжение ЦАП и линейного регулятора

Линейный регулятор необходим для согласования слаботочного выхода ЦАП с сильноточной нагрузкой (двигатель или лампы). Для соединения выхода ЦАП со входом линейного регулятора джампер JP3 нужно установить в положение "Reg".

Выход линейного регулятора (LineReg на рисунке 1.2) через перемычки (джампера) JP2 и JP1 соединяется с маломощным электродвигателем постоянного тока либо лампами накаливания (смотри рисунок 1.3).

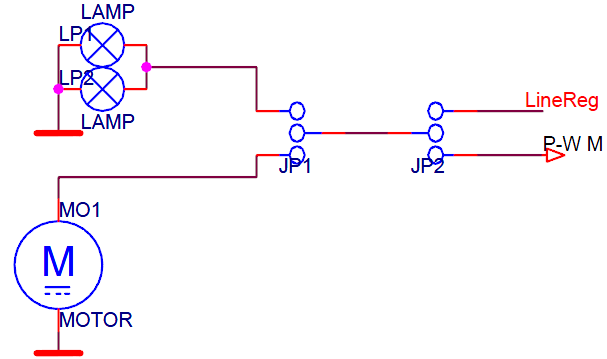


Рисунок 1.3 – Соединение линейного регулятора с электродвигателем и лампами

Из рисунка 1.3 видно, что для соединения выхода линейного регулятора с лампами или электродвигателем джампер JP2 следует установить в положение "Line". В заисимости от положения джампера JP1 может осуществляться подача напряжения с выхода линейного регулятора на лампы накаливания либо на электродвигатель. Для этого JP1 должен быть установлен в положение "Lamp" либо "Motor" соответственно.

Внимание! Все переключения джамперов в стенде и на платах расширения необходимо выполнять строго при отключенном питании стенда!

Следует отметить, что зажигание ламп накаливания и вращение электродвигателя постоянного тока начинаются лишь при достижении определённых значений напряжения на выходе ЦАП и соединённого с ним линейного регулятора. Поэтому при малых значениях цифрового кода, загружаемого в ЦАП, лампы могут не светиться, а двигатель – не вращаться.

**1.3 Порядок выполнения работы**

1 Запустите IDE AVR Studio 4.

2 Активируйте создание нового проекта посредством выбора пункта New Project из пункта Project главного меню AVR Studio 4.

3 Выберите тип создаваемого проекта: проект будет создаваться на языке ассемблер (пункт Atmel AVR Assembler) или на языке С (пункт AVR GCC).

4 В появившемся окне написать программу на языке С или ассемблер с учётом варианта задания, который указан в таблице.

Таблица 1.1 – Вариант задания

|  |  |
| --- | --- |
| № | Описание задания |
| 5 | По каждому нажатию кнопки SW15 значение напряжения, поступающее на электродвигатель постоянного тока с выхода ЦАП, увеличивается на 1В, а по каждому нажатию кнопки SW16 уменьшается на 1В. Величину напряжения, установленного на выходе ЦАП, отображать на светодиодной линейке в виде шкалы: 0В – все светодиоды потушены, 1В – горит один светодиод, ..., 5В – горит 5 светодиодов. |

5 Произвести компиляцию проекта.

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

7 Загрузить исполняемый файл проекта в микроконтроллер.

8 Визуально оценить правильность работы.

**1.4 Результаты выполнения работы**

**1.4.1 Программа, написанная на языке С, представлена ниже.**

/\*\*\* Программа к лабораторной работе №9 (стенд EV8031/AVR) \*\*\*

\*\*\*\* Работаем с лампой или моторчиком при помощи ЦАП и линейного регулятора \*\*\*

\*\*\*\* В стенд должна быть установлена плата расширения с лампой и моторчиком \*\*\*

\*\*\*\* Программа реализует плавное загорание лампы или плавный разгон моторчика \*\*\*

!!!! Внимание! Все переключения джамперов выполнять строго при отключенном питании стенда!!!!

/\*\*\* Для правильной работы программы на плате расширения должны быть установлены:

/\*\*\* - джампер JP3 в положение "Reg", чтобы соединить выход ЦАП со входом линейного регулятора;

/\*\*\* - джампер JP2 в положение "Line", чтобы соединить выход линейного регулятора с джампером JP1;

/\*\*\* - джампер JP1 в положение "Motor" либо "Lamp", чтобы подключить моторчик или лампу соответственно.

/\*\*\* Увеличение/уменьшение значения, которое выводится на ЦАП и определяет яркость лампы

\*\*\*\* или скорость вращения моторчика производится по прерываниям INT0 & INT1\*\*\*/

#define F\_CPU 7372800L //задаем частоту кварца (7,3728 МГц)

#include <avr/io.h>

#include <avr/iom8515.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#define led\_line 0xA006 //определяем адрес линейки светодиодов в стенде

#define dac\_reg 0xF000 //определяем адрес, по которому осуществляется

//доступ к записи в ЦАП на всех платах расширения

//Объявляем переменную, которая хранит значение яркости ламп или

//скорости вращения моторчика (это значение и подается на ЦАП)

volatile int currentU= 0;

//Устанавливаем указатель на адрес ЦАП

volatile unsigned char \*dac= (unsigned char\*) dac\_reg;

//Устанавливаем указатель на адрес линейки светодиодов

volatile unsigned char \*led= (unsigned char\*) led\_line;

//Объявляем вспомогательную переменную - признак того, что нам нужно делать

//Если её значение равно 0xFF, то обрабатывается прерывание IRQ0

//Если её значение равно 0x00, то обрабатывается прерывание IRQ1

//Значение этого признака изменяется в обработчике соответствующего прерывания

//массив масок загорания светодиодов

volatile unsigned char a[6] = {0b00000000, 0b00000001, 0b00000011, 0b00000111, 0b00001111, 0b00011111};

//массив значений напряжения

volatile unsigned char U[6] = {0x00, 0x33, 0x66, 0x99, 0xCC, 0xFF};

//Прерывания

//Прерывание IRQ0 (по входу INT0)

ISR(INT0\_vect){

if(currentU <5){

currentU++;

\*dac=U[currentU];

\*led=a[currentU];

}

}

//при заднем фронте

//Прерывание IRQ1 (по входу INT1)

ISR(INT1\_vect){

if (currentU > 0){

currentU--;

\*dac=U[currentU];

\*led=a[currentU];

}

}

int main(void) {

//начальная инициализация контроллера

ACSR= 1<<ACD; //отключение питания аналогового компаратора

//разрешаем работу с внешней памятью и настраиваем внешние прерывания

MCUCR= 1<<SRE | 1<< ISC11 | 0<< ISC10 | 1<<ISC01 | 0<<ISC00;

//прерывания: INT1 вызывается по низкому уровню, а INT0 - по заднему фронту

GICR= 1<<INT0 | 1<<INT1; //разрешаем внешние прерывания INT0 и INT1

//устанавливаем флаг глобального разрешения прерываний

sei(); //устанавливаем флаг глобального разрешения прерываний (будет работать только INT0 и INT1)

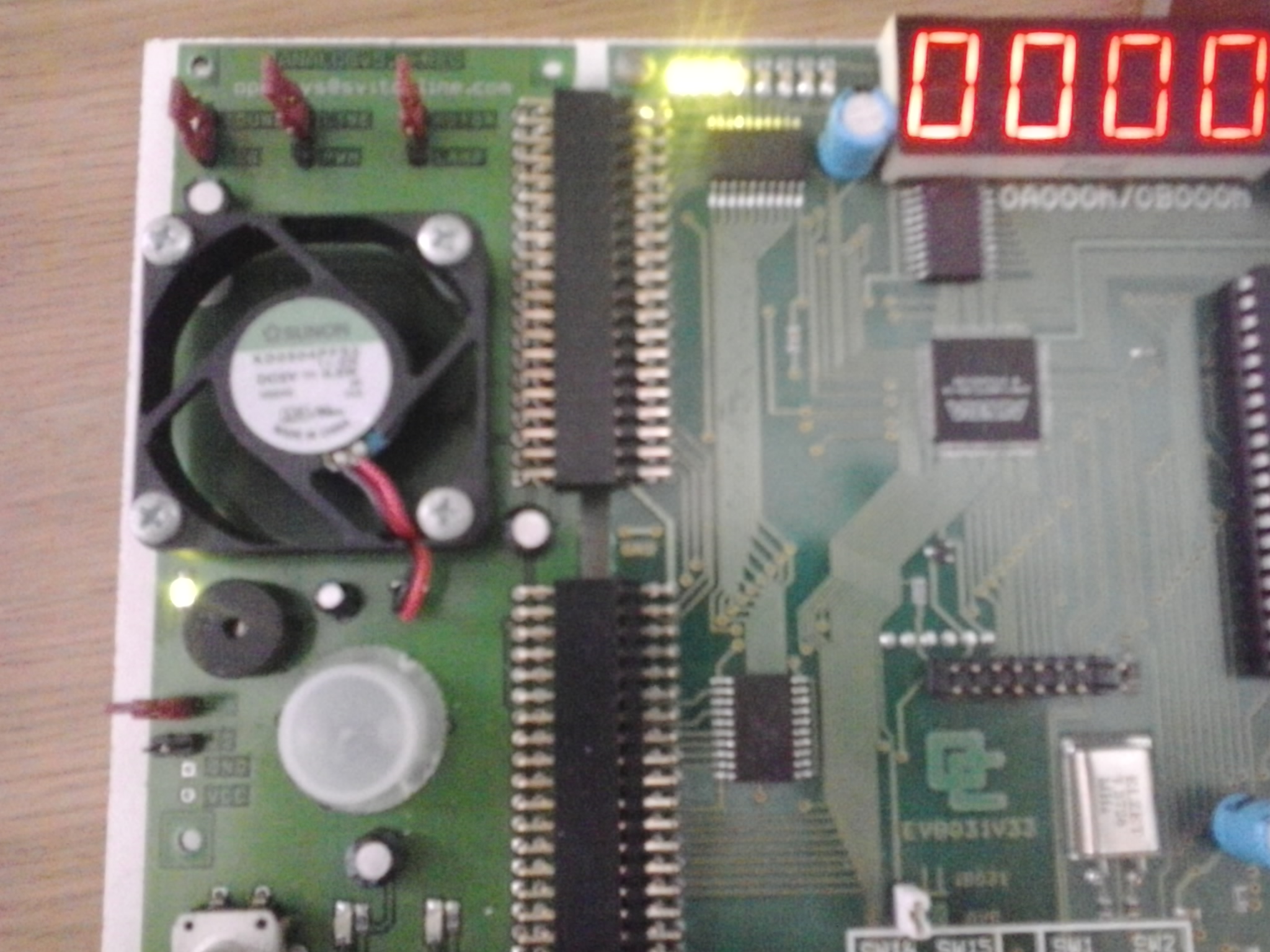
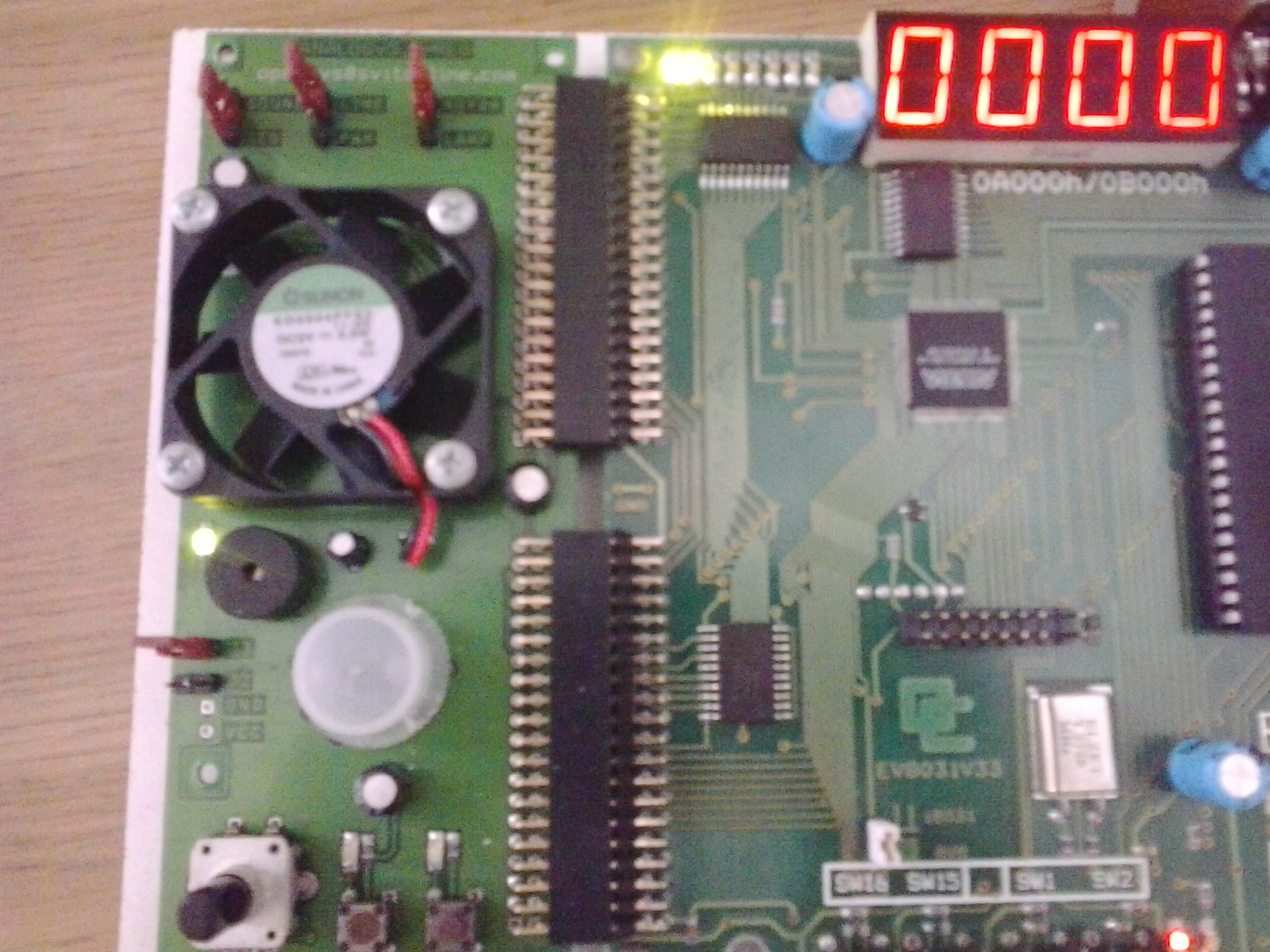
while(1) {} //бесконечный цикл (ожидаем прерывания)

return 0;

}

**1.4.2 Выполнение программы**

Рисунок 1.3 – Результат работы программы



**1.5 Особенности функционирования IDE AVR Studio 4**

AVR Studio 4 конфигурировать и обрабатывать внешние прерывания. При возникновении прерывания микроконтроллер сохраняет в стеке содержимое счётчика команд PC (адрес ячейки памяти программ, куда надо вернуться после обработки прерывания) и загружает в этот счётчик адрес вектора соответствующего прерывания. По этому адресу, как правило, находится команда безусловного перехода на подпрограммы обработки прерывания (rjmp для ATMega8515).

**1.6 Выводы**

При выполнении работы в среде AVR Studio 4на языке С была написана программа согласно варианту задания. Программа предназначена для плавного разгона/торможения электродвигателя или повышения/уменьшения яркости свечения ламп накаливания по внешним прерываниям INT0 и INT1 соответственно.