

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**КАФЕДРА САУ**

**ОТЧЕТ**  
**по Лабораторной работе №8**  
**по дисциплине «МИКРОПРОЦЕССОРНАЯ ТЕХНИКА В**  
**МЕХАТРОНИКЕ**  
**И РОБОТОТЕХНИКЕ»**  
**тема: Управление левитацией постоянного магнита в**  
**поле электромагнита**

Студенты гр. 6492

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Мурашко А.С.  
Огурецкий Д.В.  
Спорыш И.В.

Преподаватель

\_\_\_\_\_

Девяткин А.В.

Санкт-Петербург

2019

**Цель работы** — Освоение реализации системы автоматического управления неустойчивым объектом.

**Задание на лабораторную работу.** Необходимо добиться устойчивой левитации постоянного магнита в магнитном поле катушки с помощью ПД-регулятора. Реализовать возможность изменять положение магнита с помощью ползункового потенциометра.

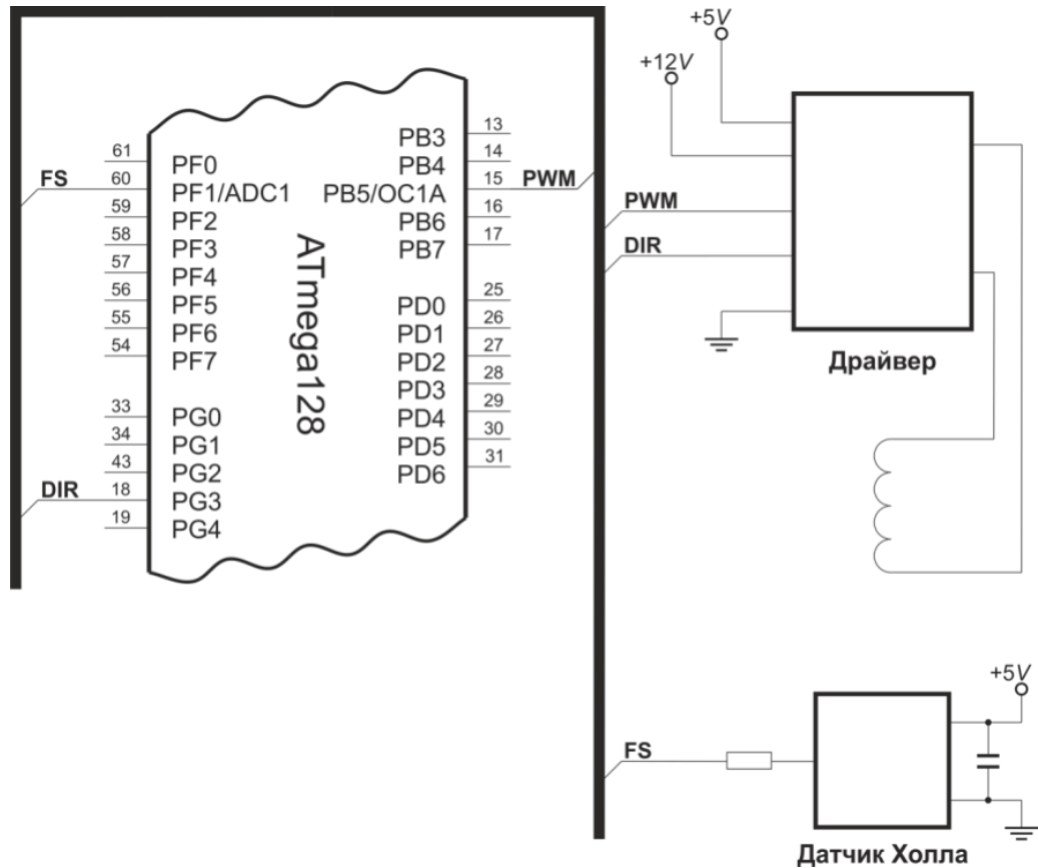


Рис. 1 Схема включения магнитного подвеса

## Выполнение работы

Обработчик прерываний от энкодера `int1Isr(void)` аналогичен и подробно описывается в 1 лабораторной работе. Функция `readAdc(unsigned char channel)` аналогична той, что использовалась в предыдущих лабораторных работах.

По нажатию на кнопку энкодера можно менять параметр для настройки. \*

Светодиод индицирует параметр, который выбран в данный момент:  
green — Kd

red — Kp

blue — control mode

\* изначально kp=3 kd=10

Нажатие кнопки вызывает обработчик прерываний NameIsr()

```
1 interrupt [EXT_INT2] void NameIsr(void)
2 {
3     if (BIT_IS_CLEAR(PIND, 2))
4     {
5         button++;
6         switch (button)
7         {
8             case 1: //настройка следующего коэффициента
9                 f_Kp=1;
10                count=Kp;
11                break;
12             case 2: //начать управление
13                 flag_control=1;
14                 f_Kp=0;
15                 PORTE = _BV(3);
16                break;
17             case 3: //вернуться в режим настройки
18                 PORTE=0; //погасить светодиоды
19                 button=0;
20                 count=Kd;
21                 flag_control=0;
22                break;
23         }
24     }
25 }
```

## Описание main()

Работа программы подробно описана на блок-схеме.

## Код

```
1 unsigned int Kp=3;
2 unsigned int Kd =0;
3 unsigned int count = 10; //счетчик суммы энкодера
4 char button = 0; //количество нажатий
5 char flag_control = 0; //переход у кправлению
6
7 char f_Kp = 0; //коэффициент для настройки и отображения
8
9 void main(void)
10 {
11     int field; // Значение с датчика Холла
12     unsigned long ref; // Задание
13     int error = 0; // Ошибка управления
14     int control; // Сигнал управления
15     int error_last=0; //Ошибка управления последнее
16     init_segments();
17     #asm("sei"); //разрешение прерываний
18     // внешнее прерывание по фронту на входе INT1
19     EICRA |= _BV(ISC11) | _BV(ISC10) ;//на фронт
20     EIMSK |= _BV(INT1);
```

```

21 // Инициализация портов ввода/вывода
22 DDRB |= _BV(5) ; // ШИМ управления
23 DDRG |= _BV(3); // направление тока
24 //прерывание по нажатию на кнопку
25 EICRA |= _BV(ISC21) ;
26 EIMSK |= _BV(INT2) ;
27 DDRE |= _BV(5) | _BV (4) | _BV(3) ;
28 // Инициализация АЦП
29 ADCSRA = _BV(ADEN) | _BV(ADPS2) | _BV(ADPS1) | _BV(ADPS0);
30 //freq=11.0529/128=0,08635078125 MHz T=11.58 us/tick
31 // Инициализация таймера 1
32 // Быстрая ШИМ, 8 бит, без предделителя
33 TCCR1A = _BV(COM1A1) | _BV(WGM10);
34 TCCR1B = _BV(WGM12) | _BV(CS10);
35 while(1)
36 {
37     if(flag_control)
38     {
39         //управления
40         // Получение и фильтрация сигнала с датчика
41         // магнитного поля
42         field = (readAdc(1) + readAdc(1) +
43         readAdc(1) + readAdc(1) + readAdc(1)) / 5;
44
45         ref = readAdc(3)*182;
46         ref = ref/1023+508; // Получение задания с ручки потенциомера
47         if(field<=2) //магнит лежит
48         {
49             ref=559;//подбрасываем магнит, чтобы он нележал
50         }
51
52         // Расчёт ошибки управления
53         error = ref - field ;
54         // Расчёт сигнала управления
55         control = error/Kp+(error-error_last)*Kd;
56         error_last=error;
57
58         if(field>=290) { control=40;} //прилипание, нужно "отпу-
59 стить магнит"
60
61         // Ограничение сигнала управления
62         if(control > 255)
63         {
64             control = 255;
65         }
66         if(control < -255)
67         {
68             control = -255;
69         }
70
71         // Установка сигнала управления
72         if(control >= 0)
73         {
74             PORTG &= ~_BV(3); //установка положительного тока
75             OCR1A1 = (unsigned char)control;
76         }
77         else
78         {
79             PORTG |= _BV(3); //установка отрицательного тока

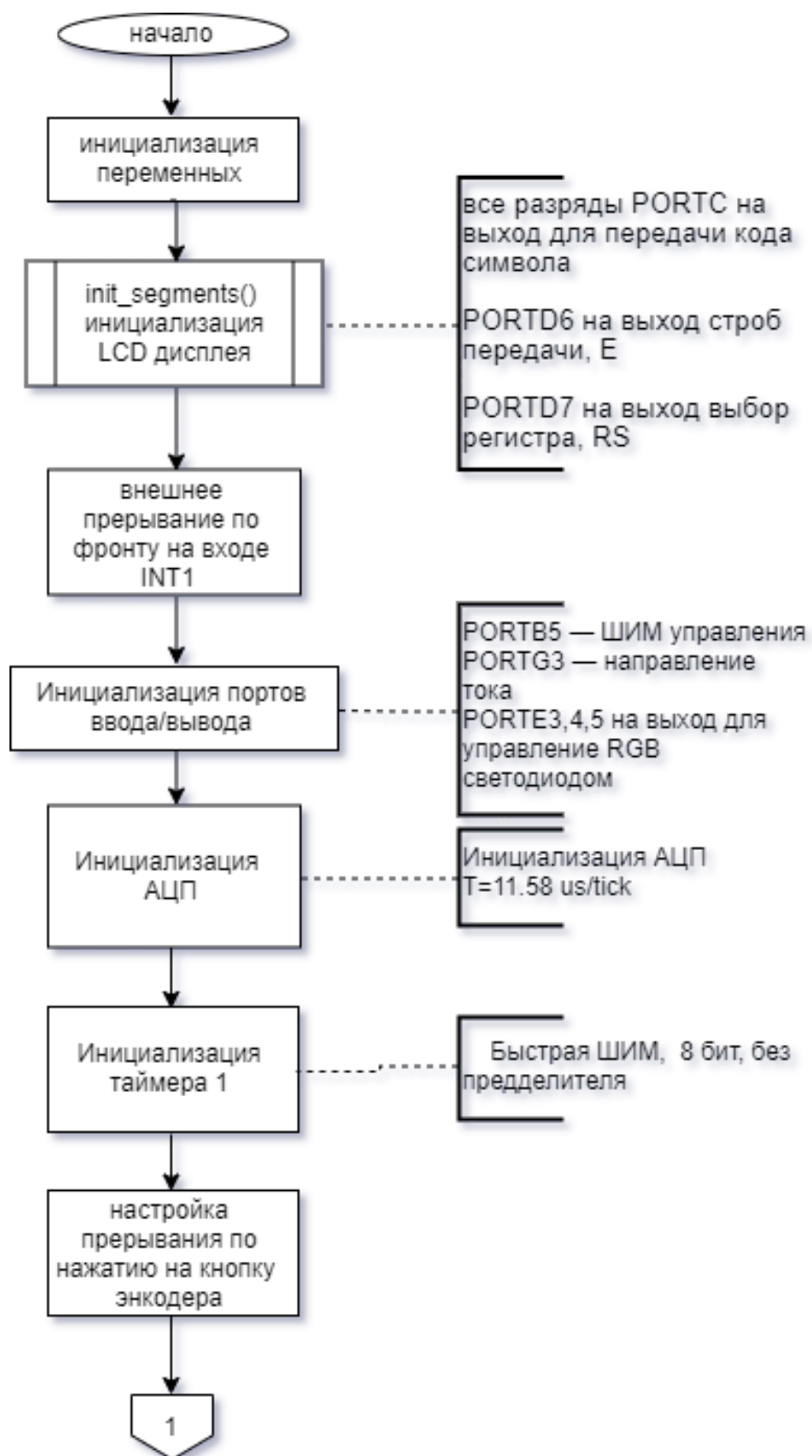
```

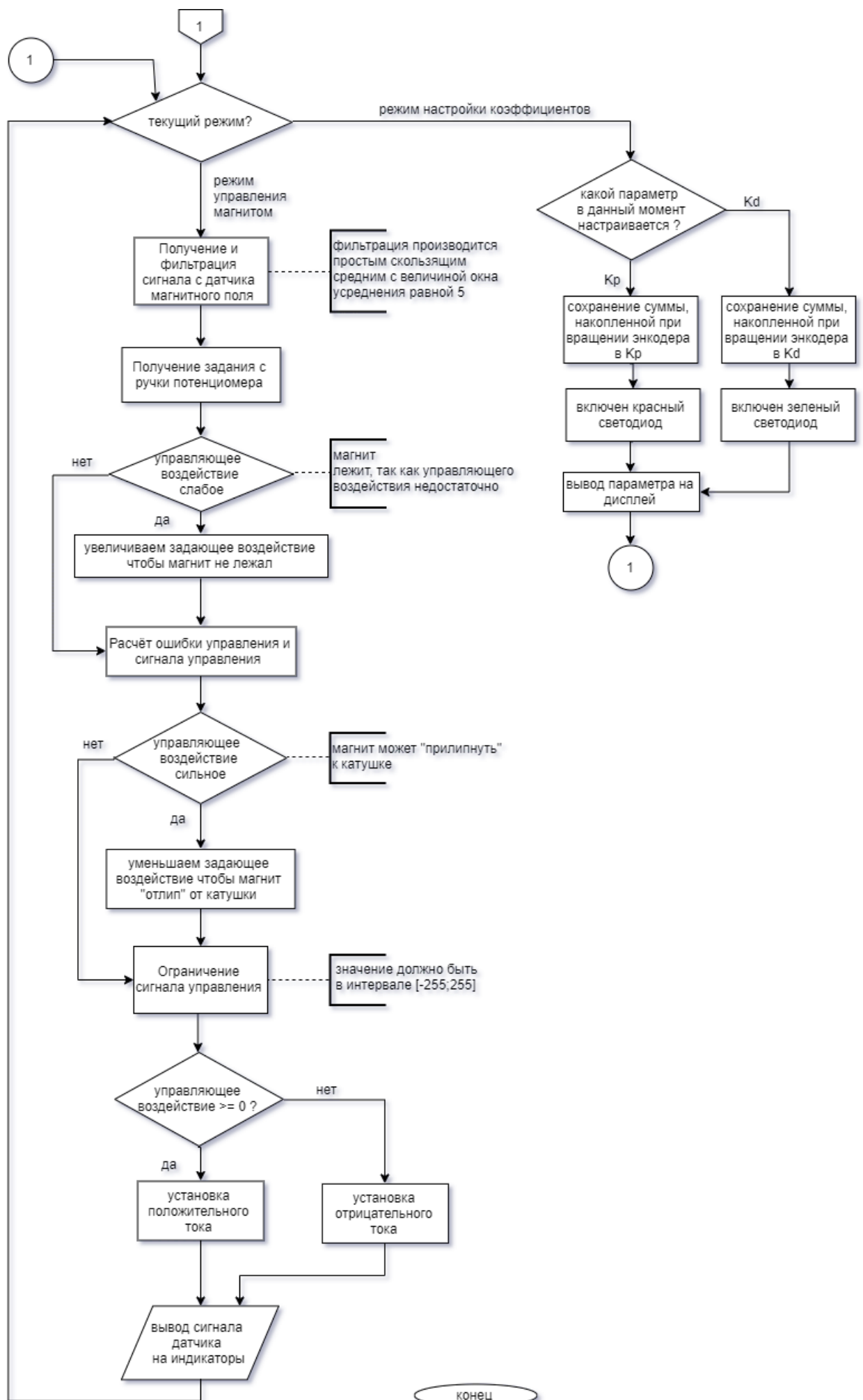
```

80         OCR1AL = (unsigned char) (-control);
81     }
82     indic_int( field );
83 }else //настройка коэффициентов
84 {
85     if(f_Kp){ Kp=count; }else{ Kd=count; };
86     BitClr(PORTE,4);
87     BitClr(PORTE,5);
88     BitSet(PORTE, (f_Kp ? 5 : 4 ));
89     indic_uint( f_Kp ? Kp : Kd );
90 }
    }
}

```

## Блок-схема main()





Вывод: с помощью микроконтроллера можно управлять левитацией магнита. Таким образом можно быстро создать систему управления с требуемым регулятором, например систему управления левитацией магнита. В данной работе коэффициенты подбирались экспериментально, но с использованием, если необходимо более точное управление, то необходимо знать математическую модель объекта управления.