

Министерство науки и высшего образования Российской Федерации
Ульяновский государственный технический университет

Лабораторная работа № 4 по предмету
«Алгоритмические и аппаратные средства обработки информации»

РЕАЛИЗАЦИЯ НЕРЕКУРСИВНЫХ КИХ-ФИЛЬТРОВ НА БАЗЕ
МИКРОКОНТРОЛЛЕРОВ

(Название лабораторной работы)

Учебная группа ИСТМД-11

	ФИО	Дата	Подпись
Студент	Шаблыгин В.В.		
Преподаватель	Сазонов С.Н.		

Ульяновск, 2022

ЦЕЛЬ РАБОТЫ:

Изучить методику построения цифровых фильтров на основе микроконтроллеров.

Задача:

1. В программном пакете PROTEUS собрать схему, представленную на рисунке 4.8.
2. По варианту задания установить параметры синусоидальных генераторов (амплитуды U_n и U_v выбрать так, чтобы выходное напряжение U_2 не превышало 2,5 В).
3. Реализовать программу цифрового фильтр скользящего среднего, а также программы цифровых КИХ-фильтров с прямоугольным окном, окном Хэмминга, окном Блэкмана и окном Кайзера.
4. Вывести графики выходных напряжений для всех типов фильтров в режимах аналогового и Фурье анализа и сравнить между собой.

ОБОРУДОВАНИЕ И ПРИНАДЛЕЖНОСТИ:

Программный пакет PROTEUS.

ВЫПОЛНЕНИЕ РАБОТЫ

Вариант № 8

Исходные данные:

$$F_n = 15 \text{ Гц}$$

$$F_v = 87 \text{ Гц}$$

$$U_n/U_v = 6$$

1. Алгоритм цифрового фильтра скользящего усреднения:

$$D_{acc} = D_{acc} + D_{in} - D_{out}$$

$$D_{out} = D_{acc} / K$$

где:

D_{out} - выходное значение фильтра,

D_{in} - входное значение фильтра,

K - постоянный коэффициент, который рассчитывается по формуле:

$$K = T \times SPS$$

где:

T - это постоянная времени фильтра,

SPS - частота дискретизации АЦП.

Тактовая частота модуля АЦП равна отношению тактовой частоты контроллера к коэффициенту внутреннего делителя:

$$F_{adc} = F_{cpu}/Pre = 8000000/128 = 62500 \text{ Гц}$$

Одно преобразование занимает 13 тактов (по даташиту), тогда частота дискретизации:

$$F_s = F_{adc}/n = 62500/13 = \sim 4800 \text{ Гц}$$

Постоянную фильтра найдем из формулы:

$$f = 1 / 2\pi RC$$

Для частоты 87 Гц, постоянная времени фильтра:

$$T = 1/2\pi f = 0.0018 \text{ с}$$

1. Метод скользящего среднего.

Листинг программы:

```
1  #include <avr/io.h>
2  #include <stdint.h>
3  #define SPS 4800UL // частота дискретизации
4  #define Trc 0.0018f // постоянная времени фильтра
5  #define K (SPS*Trc) // постоянный коэффициент
6
7  void ADC_INIT() // конфигурирование АЦП
8  {
9      ADCSRA |= (1<<7); // Бит ADEN=1 включение АЦП
10     ADCSRA |= (1<<2) | (1<<1) | (1<<0); // ADPS0..2=111, делитель - CLK/128
11     ADCSRA |= (1<<5); // ADSCF=1, запуск ФЦП по прерыванию
12     SFIOR &= ~(1<<7) & (1<<6) & ~(1<<0); // ADTS2..0=010 - запуск от внеш. прерывания
13     ADMUX &= (1<<7) | (1<<6); // выбор ИОН, 11 - внутренний ИОН 1.1В
14     ADMUX &= (1<<0); // выбор 0-го входа мультимплексора
15     ADMUX &= (1<<5); // выравнивание рез-та по левому краю
16 }
17
18 int main(void)
19 {
20     ADC_INIT();
21     ADCSRA |= (1<<6); // запуск преобразования
22     DDRD = 0xFF; // режим регистра D - выход
23     DDRC = 0xFF; // режим регистра C - выход
24     uint32_t Dacc = 0; // устанавливаем начальное среднее значение
25     uint16_t Dout = 0; // выходное значение фильтра
26
27     while (1)
28     {
29         if (ADCSRA & (1<<4)) // флаг прерывания от АЦП
30         {
31             // ADCL и ADCH - верх-й и ниж-й р-ры с результатом преобразования АЦП
32             uint8_t lowADC = ADCL;
33             uint16_t Din = ADCH << 8 | lowADC;
34             Dacc = Dacc + Din - Dout; // расчет скользящего среднего
35             Dout = Dacc / (uint32_t)K;
36             PORTD = (uint8_t)Dout; // результат записываем в р-р порта D (мл.байт)
37             PORTC = (uint8_t)(Dout >> 8); // в р-р порта C записываем ст. байт
38             ADCSRA |= (1<<4); // Бит ADIF - флаг прерывания от АЦП
39         }
40     }
41 }
```

Для проверки, нужно собрать схему, и загрузить в виртуальный микроконтроллер, полученную программу.

2. В Scilab рассчитаем коэффициенты фильтров 4-го порядка с прямоугольным окном, окном Хэмминга, окном Блэкмана и окном Кайзера.

```
1 n=4
2 //Rectangle
3 wr=1:n;
4 for i=0:n-1
5     wr(i+1)=0.54-0.46*cos((2*pi*i)/(n-1));
6 else
7     wr(i+1)=1;
8 end
9 end
10 disp('Прямоугольное окно')
11 disp(wr)
12
13 //hemmig
14 [wft,wfm,fr]=wfir("lp", n, [-.015-.0], "hm", [0-0]);
15 disp('Hemming:')
16 disp(wft);
17
18 //kaiser
19 [wft,wfm,fr]=wfir("lp", n, [-.015-.0], "kr", [0-0]);
20 disp('Kaiser:')
21 disp(wft);
22
23 //blackman
24 wb=1:n;
25 for i=0:n-1
26     wb(i+1)=0.42-0.5*cos((2*pi*i)/(n-1))-.08*cos(4*pi*i/(n-1));
27 end
28 disp('Blackman:')
29 disp(wb)
```

Результаты расчетов:

"Прямоугольное окно:"

0.08 0.77 0.77 0.08

"Hemming:"

0.002392 0.0230915 0.0230915 0.002392

"Kaiser:"

0.0299002 0.0299889 0.0299889 0.0299002

"Blackman:"

-1.388D-17 0.63 0.63 -1.388D-17

Далее подставляем полученные коэффициенты в программу:

```

19 int main(void)
20 {
21     ADC_INIT();
22     ADCSRA |= (1<<6); // запуск преобразования
23     DDRD = 0xFF; // режим регистра D - выход
24     DDRC = 0xFF; // режим регистра C - выход
25     uint16_t Dout = 0; // выходное значение фильтра
26     // значения коэффициентов
27     static const float b0 = 0.08;
28     static const float b1 = 0.77;
29     static const float b2 = 0.77;
30     static const float b3 = 0.08;
31     // линия задержки
32     uint16_t x3 = 0.0;
33     uint16_t x2 = 0.0;
34     uint16_t x1 = 0.0;
35     uint16_t x0 = 0.0;
36
37     while (1)
38     {
39         if (ADCSRA & (1<<4)) // флаг прерывания от АЦП
40         {
41             // ADCL и ADCH - верх-й и нижн.р-ры с рез-том преобразования АЦП
42             uint8_t lowADC = ADCL;
43             uint16_t Din = ADCH << 8 | lowADC;
44             // двигаем регистр
45             x3=x2;
46             x2=x1;
47             x1=x0;
48             x0=lowADC;
49             // Считаем "отфильтрованное" значение
50             Dout = x0*b0 + x1*b1 + x2*b2 + x3*b3;
51             PORTD = (uint8_t)Dout; //результат записываем в р-р порта D (мл.байт)
52             PORTC = (uint8_t)(Dout >> 8); // в р-р порта C записываем ст. байт
53             ADCSRA |= (1<<4); // Бит ADIF - флаг прерывания от АЦП
54         }
55     }
56 }

```

Поочередно записываем полученный код в микроконтроллер и запускаем симуляцию:

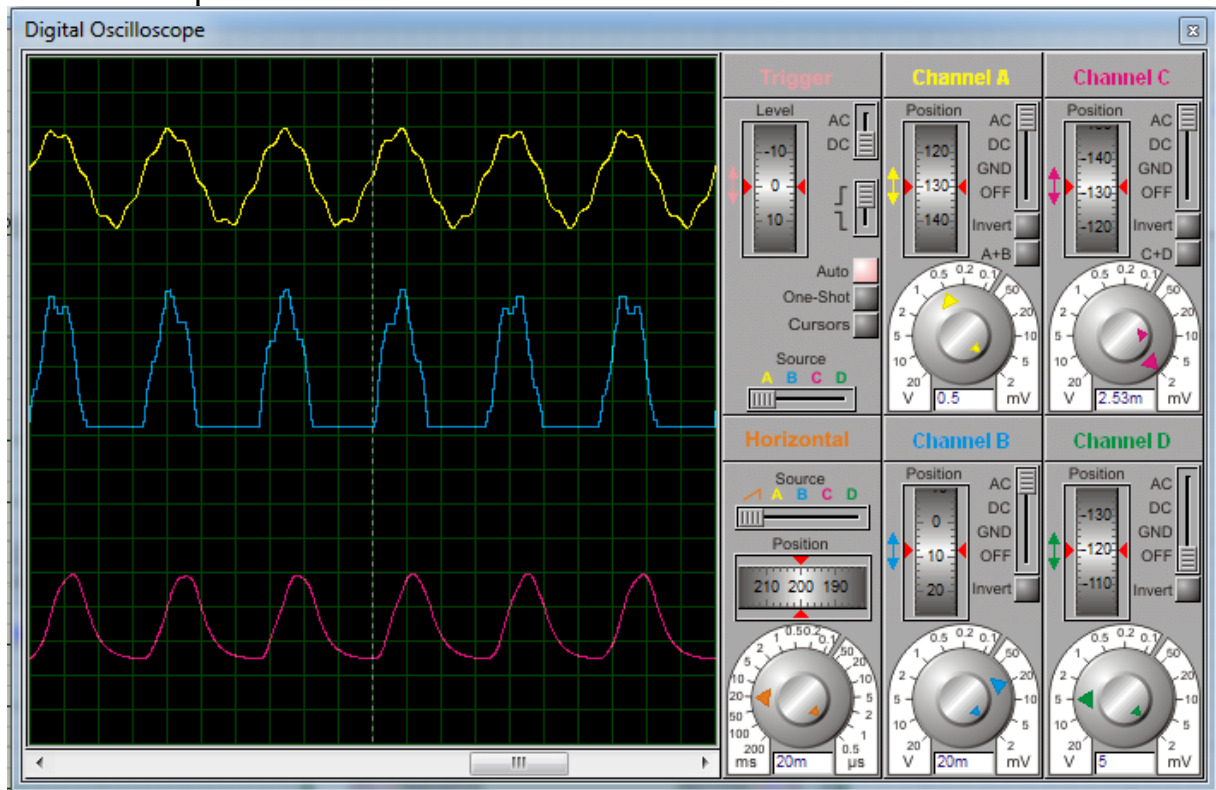
The screenshot displays a digital oscilloscope interface. On the left, a large screen shows three waveforms: a yellow sine wave at the top, a blue square wave in the middle, and a pink sine wave at the bottom. A vertical dashed white line is positioned in the center of the screen. To the right of the screen are six control panels arranged in a 3x2 grid. The top row contains the 'Trigger' panel and panels for 'Channel A' and 'Channel C'. The bottom row contains the 'Horizontal' panel and panels for 'Channel B' and 'Channel D'. Each panel includes various controls such as level, position, AC/DC coupling, source selection, and range/delay settings. The 'Trigger' panel has a 'Level' knob set to 0 and an 'Auto' button. The 'Horizontal' panel has a 'Position' knob set to 210 and a 'Source' selector set to A. The 'Channel A' panel has a 'Position' knob set to 120 and a 'Range' knob set to 0.5 mV. The 'Channel C' panel has a 'Position' knob set to -130 and a 'Range' knob set to 25.33 mV. The 'Channel B' panel has a 'Position' knob set to 10 and a 'Range' knob set to 0.5 mV. The 'Channel D' panel has a 'Position' knob set to -120 and a 'Range' knob set to 5 mV. The bottom of the screen features a time scale control set to 20 ms and a unit selector set to μs.

The screenshot displays a digital oscilloscope interface. The main display area on the left shows three waveforms: a yellow sine wave at the top, a blue square wave in the middle, and a pink sine wave at the bottom. A vertical dashed white line is positioned in the center of the grid. The right side of the interface features several control panels:

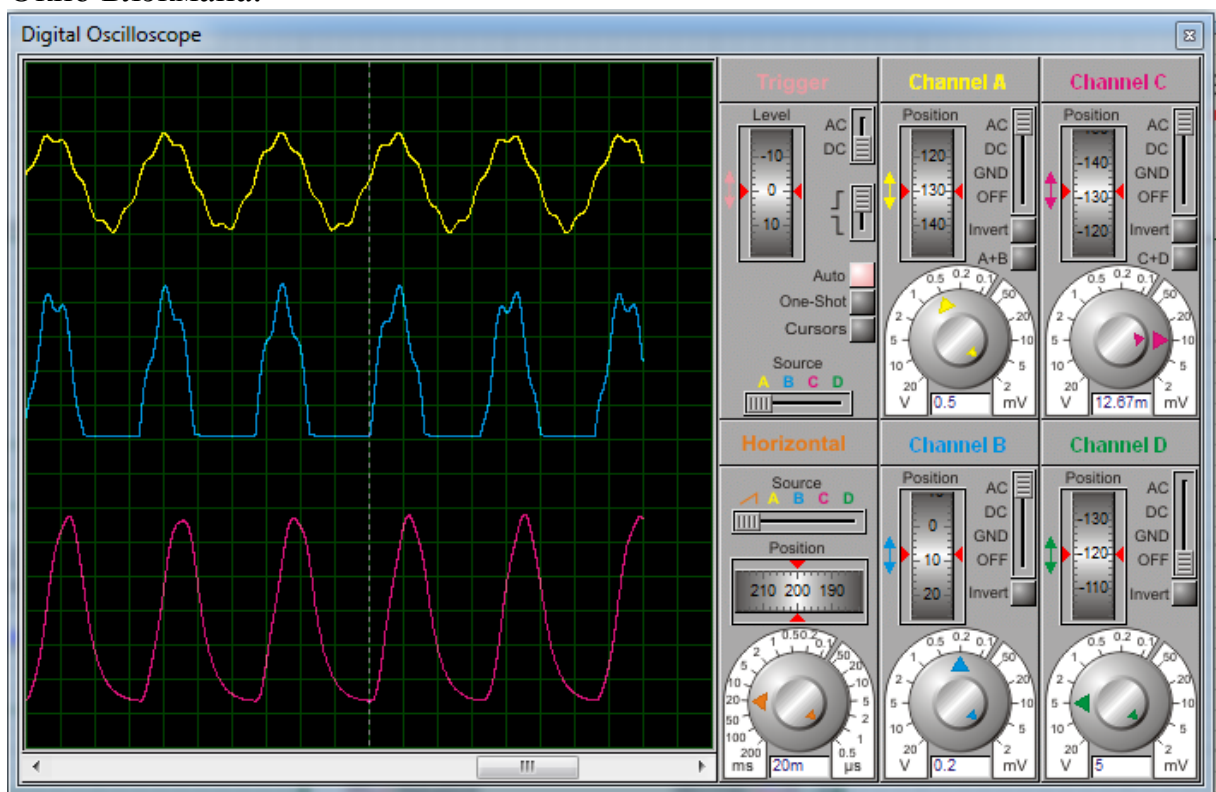
- Trigger:** Includes a Level slider (set to 0), AC/DC coupling, a Source selector (set to A), and a Position slider (set to 210).
- Channel A:** Includes a Position slider (set to 130), AC/DC coupling, and a Source selector (set to A).
- Channel B:** Includes a Position slider (set to 10), AC/DC coupling, and a Source selector (set to B).
- Channel C:** Includes a Position slider (set to -130), AC/DC coupling, and a Source selector (set to C).
- Channel D:** Includes a Position slider (set to -130), AC/DC coupling, and a Source selector (set to D).

Each channel panel also features a vertical scale knob and a horizontal scale knob. The bottom of the interface includes a time base control (set to 20ms) and a horizontal position slider.

Окно Кайзера:



Окно Блэкмана:



Вывод: в результате данной работы был рассчитан и смоделирован цифровой фильтр низких частот.