

Центр «Солярис» при Ульяновском государственном университете

Исследовательская Творческая Группа «Солярис»

МБОУ «Лицей физики, математики, информатики №40» при УлГУ



# Блок питания и управления вибродвигателем

**Автор:**

**Муравьёв Евгений Сергеевич**, 9«А» класс.

**Руководитель:**

**Иванов Илья Петрович**,

руководитель ИТГ «Солярис», директор Центра «Солярис» при УлГУ,

научный сотрудник НИТИ им.С.П.Капицы УлГУ,

педагог дополнительного образования высшей категории.

**E-mail:** [SolarisGroup@mail.ru](mailto:SolarisGroup@mail.ru)

Ульяновск, 2015

Данный проект вытекает из задач другого проекта Соляриса - «Исследование аэродинамических вибродвигателей и перспективы создания вибролёта». Для исследования межтактовой разности скоростей движения аэродинамических вибродвигателей нами было принято решение о создании **электрического вибродвигателя (ДЭВ)** особой конструкции с активным прямым ходом штока («вниз») и пассивным (под действием возвратной пружины) обратным ходом, который должен запитываться от 2-х-киловаттного **блока питания** и управляться **силовым полевым транзистором**, в свою очередь, управляемого **микроконтроллером**. Для этого нами был создан вибродвигатель БВД-2 (рис.1, справа) и **блок питания и управления (БПУ)** к нему (рис.1, слева), который представлен в данной работе.



**Рис.1.** Внешний вид БПУ (слева) и БВД-2 (справа)

**БПУ** состоит из объединённых в одном корпусе **блока питания (БП)** и **блока управления (БУ)**. Последний пространственно отделён внутри корпуса БПУ от БП и также выделен на передней панели (см. на рис.1 - в правом нижнем углу панели).

**Блок питания (БП)** состоит из следующих элементов (см. схему на рис.2):

1. ЛАТРа, выдающего напряжение от 0 до 250-260 В мощностью 2 кВт;
2. диодного моста КВРС2510 (25А 1000В);
3. батареи конденсаторов (6000 мкф);
4. автомата на 20А в цепи питания прибора от сети (~220 В) - на передней панели;

# Блок питания и управления (БПУ)

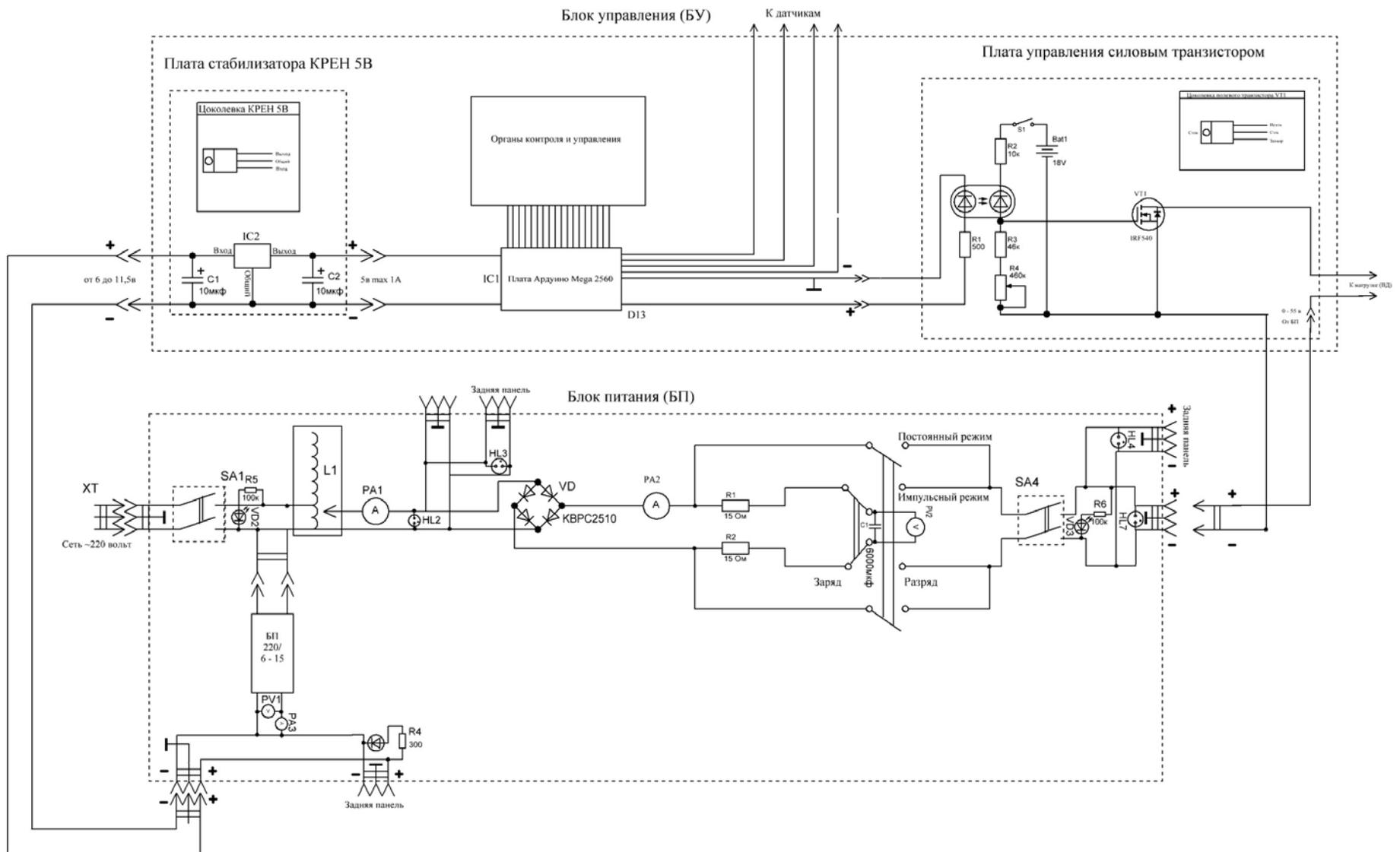


Рис.2. Схема БПУ

5. **автомата** на 16А на выходе цепи постоянного высокого напряжения (до 350 - 360В) - на передней панели;
6. **управляющих тумблеров** на передней панели;
7. **светодиодных и неоновых индикаторов**, выведенных на переднюю панель;
8. **низковольтного источника питания** с регулировкой от 6 до 15 В;
9. нескольких **вольтметров и амперметров**, контролирующих различные цепи прибора - на передней панели.

**Низкое постоянное напряжение БП может ступенчато регулироваться** ручкой на передней панели в пределах **от 6 до 15 вольт**.

**Высокое переменное напряжение БП может плавно регулироваться** рукояткой ЛАТРа в интервале **от 0 до 250-260 вольт**, к которой имеется доступ через окошко в крышке прибора (см. рис.1). Оно выпрямляется диодным мостом и сглаживается батареей конденсаторов, формируя **постоянное высокое напряжение от 0 до 350-360 вольт**. В БП предусмотрены отключаемые управляющими тумблерами **мощные керамические резисторы** по 15 Ом для сглаживания броска тока при включении прибора, а также защитные автоматы на 20 и 16 ампер в цепях, соответственно, питания и выхода постоянного высокого напряжения. **Два управляющих тумблера** обеспечивают **несколько режимов работы** прибора, в частности **постоянный и импульсный**. В **постоянном режиме** выход диодного моста постоянно соединён с батареей конденсаторов и выходом постоянного высокого напряжения БП. В **импульсном режиме** можно осуществлять заряд батареи конденсаторов от ЛАТРа и диодного моста до нужного напряжения, отключение батареи от диодного моста и разряд её через блок управления через нагрузку. Контролируя напряжение на батарее до и после разряда, и зная ёмкость батареи (6000 мкф), можно вычислить **энергию разряда** по формуле:

$$E_{разр} = \frac{C}{2} (U_{до}^2 - U_{после}^2)$$

Это необходимо, к примеру, для определения **КПД** двигателя,

являющегося нагрузкой.

Низковольтное постоянное, высоковольтное переменное и постоянное напряжения от БП, помимо подачи на блок управления внутри прибора, также **выведены на разъёмы на заднюю панель прибора.**

**Блок управления (БУ)** состоит из (см. схему на рис.2):

1. **платы Arduino Mega 2560** с микроконтроллером ATmega2560 [4];
2. **платы стабилизатора** на основе КРЕН 5В;
3. **платы управления силовым транзистором;**
4. **органов контроля и управления** - кнопок, светодиодных индикаторов, переменного резистора, выведенных на переднюю панель;
5. **выходных разъёмов** на правой и задней панелях.

**Плата Arduino Mega 2560 [4]** запитывается через **плату стабилизатора КРЕН** от **низкого постоянного напряжения БП.**

Высокое постоянное напряжение БП через **силовой полевой транзистор**, выходит на специальные разъёмы на задней панели для подключения нагрузки (ВД). Транзистор управляется платой **Arduino Mega 2560 [4]** посредством написанной нами специальной программы. Высоковольтная цепь транзистора и низковольтная цепь **Arduino Mega 2560 гальванически развязаны** посредством созданной нами **оптопары**. Управляющее напряжение на затвор транзистора формируется настраиваемым делителем напряжения (в который входит включённый в обратном направлении фотодиод ФД-256 оптопары), получаемого от автономного источника в 18 вольт (2 «Кроны»). При длительном бездействии прибора, «Кроны» можно отключить переключателем. Все перечисленные элементы составляют **плату управления силовым транзистором.**

Конструкция прибора позволяет **увеличивать количество плат управления силовыми транзисторами** для осуществления разнообразных способов управления вибродвигателем (вибродвигателями).

В процессе создания БПУ мы столкнулись с тем, что **силовые полевые транзисторы выходили из строя**, хотя их характеристики, казалось бы, соответствовали требованиям по напряжению и току. Мы выяснили, что это происходило из-за того, что при коммутации индуктивной нагрузки возникали большие напряжения вследствие **самоиндукции**, а возможно и **резонанса**, которые выводили из строя транзисторы. В настоящее время запас по напряжению у транзисторов в приборе составляет **от 2,5 до 5 раз**.

На переднюю панель выведены **органы контроля и управления БУ**: кнопки, светодиодные индикаторы, а также переменный резистор для плавной регулировки тех или иных параметров выходного тока (плавная регулировка осуществляется **программным путём**, т.е., через микроконтроллер, а переменный резистор заведён по схеме переменного делителя напряжения на аналоговый вход (АЦП) микроконтроллера).

В боковой (правой) стенке прибора предусмотрен **разъём** (типа «VGA») для подключения **датчиков** от вибродвигателя для осуществления обратной связи (на рисунке 1 он не виден).

Все нагревающиеся элементы прибора (силовой транзистор, КРЕН, керамические резисторы, диодный мост) поставлены на **радиаторы**, а некоторые снабжены ещё и **кулерами**, питающимися от низковольтного постоянного выхода БП.

БПУ обеспечивает **импульсное управление вибродвигателем** путем формирования силовых импульсов различной **частоты (от 1 до 60 Гц)** и **скважности (от 1 до 10)** с плавной регулировкой данных параметров. Коммутируемое прибором напряжение составляет **от 0 до 200 В**, а коммутируемый ток - **до 10 А**. Программой, загруженной в настоящее время в микроконтроллер (см. Приложение), предусматривается **5 режимов работы** прибора:

- 1) простое **включение и выключение** тока пусковой кнопкой;

2) подача тока с определённой, плавно регулируемой **частотой от 1 до 50 Гц** при **фиксированной скважности 2**;

3) подача тока с определённой, плавно регулируемой, **скважностью от 1 до 10** при **фиксированной частоте 60 Гц**;

4) подача тока с **плавно регулируемой частотой от 1 до 50 Гц** при ранее **плавно выбранной скважности 1 до 10**;

5) подача тока с **плавно регулируемой скважностью 1 до 10** при ранее **плавно выбранной частоте от 1 до 50 Гц**.

Прибор **практически закончен** (хотя и **допускает модификации**), он, в основном, соответствует требованиям технического задания, и уже **использовался нами по назначению** (в исследовательских целях в рамках проекта «Вибролёт»).

## **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Ревич Ю. В. Занимательная электроника. СПб.: БХВ-Петербург 2005 г. \\  
<http://madeelectronics.ru/book/radiolyubitelyam/2010-02-15-12-55-45-632.htm>
2. Николаенко М.Н. Самоучитель по радиоэлектронике / М.Н. Николаенко. – М.:ИТ Пресс, 2006. – 224 с.
3. Никитин В.А. Книга начинающего радиолобителя / В.А. Никитин. – М.:ИТ Пресс, 2007. – 384с.
4. Arduino Mega 2560 \\  
<http://arduino.ru/Hardware/ArduinoBoardMega2560>
5. Генрих Альтшуллер: Найти идею. Введение в ТРИЗ - теорию решения изобретательских задач, Альпина Паблишер, 2015 г. \\  
<http://www.labirint.ru/books/139105/>

# ПРИЛОЖЕНИЕ

## Программа управления БПУ

```
#define BLUE_BUTTON 41 // синяя кнопка
#define BIG_RED_BUTTON 39 // большая красная кнопка

//*****

#define LED_BLUE_1 37 // синие светодиоды (снизу вверх)
#define LED_BLUE_2 53
#define LED_BLUE_3 51
#define LED_BLUE_4 49
#define LED_BLUE_5 47

#define LED_RED_1 52 // красные светодиоды - полосы (слева направо)
#define LED_RED_2 50
#define LED_RED_3 48
#define LED_RED_4 46
#define LED_RED_5 42

#define LED_RED_CIRCUIT 40 // контур полоскового индикатора

#define LED_BIG_BUTTON 35 // светодиод "большая кнопка"

#define BEEPER 45 // пищалка

#define TRANSISTOR_1 13 // управление 1-м силовым транзистором

//*****

#define VAR_RESISTOR A0 // переменный резистор

//*****

float T; // период следования управляющих импульсов, мкс
float S; // скважность управляющих импульсов
float tau; // длительность управляющих импульсов, мкс

long time, time_pred = 0;

int reg; // номер режима работы
float Uvr; // напряжение на средней ножке переменного резистора (от 0 до 1023)

bool red_button_pressed; // индикатор нажатия большой красной кнопки
bool blue_button_pressed; // индикатор нажатия синей кнопки
bool work; // индикатор работы
```

```

//*****

void setup()
{

//*****

pinMode(BLUE_BUTTON, INPUT);
pinMode(BIG_RED_BUTTON, INPUT);

//*****

pinMode(LED_BLUE_1, OUTPUT);
pinMode(LED_BLUE_2, OUTPUT);
pinMode(LED_BLUE_3, OUTPUT);
pinMode(LED_BLUE_4, OUTPUT);
pinMode(LED_BLUE_5, OUTPUT);

pinMode(LED_RED_1, OUTPUT);
pinMode(LED_RED_2, OUTPUT);
pinMode(LED_RED_3, OUTPUT);
pinMode(LED_RED_4, OUTPUT);
pinMode(LED_RED_5, OUTPUT);
pinMode(LED_RED_CIRCUIT, OUTPUT);

pinMode(LED_BIG_BUTTON, OUTPUT);

pinMode(BEEPER, OUTPUT);

pinMode(TRANSISTOR_1, OUTPUT);

//*****
// включение на 1 секунду и выключение всех индикаторов

digitalWrite(LED_BLUE_1, HIGH);
digitalWrite(LED_BLUE_2, HIGH);
digitalWrite(LED_BLUE_3, HIGH);
digitalWrite(LED_BLUE_4, HIGH);
digitalWrite(LED_BLUE_5, HIGH);

digitalWrite(LED_RED_1, LOW);
digitalWrite(LED_RED_2, LOW);
digitalWrite(LED_RED_3, LOW);
digitalWrite(LED_RED_4, LOW);
digitalWrite(LED_RED_5, LOW);
digitalWrite(LED_RED_CIRCUIT, LOW);

digitalWrite(LED_BIG_BUTTON, HIGH);

digitalWrite(BEEPER, HIGH);

delay(1000);

```

```

digitalWrite(LED_BLUE_1, LOW);
digitalWrite(LED_BLUE_2, LOW);
digitalWrite(LED_BLUE_3, LOW);
digitalWrite(LED_BLUE_4, LOW);
digitalWrite(LED_BLUE_5, LOW);

digitalWrite(LED_RED_1, HIGH);
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);
digitalWrite(LED_RED_CIRCUIT, HIGH);

digitalWrite(LED_BIG_BUTTON, LOW);

digitalWrite(BEEPER, LOW);

//delay(1000);

//*****

reg = 1;

//*****

}

//*****

void loop()
{
while(blue_button_pressed) // ожидание отжатия синей кнопки
  if (digitalRead (BLUE_BUTTON) == LOW) blue_button_pressed = false;

//*****
// выбор номера режима

digitalWrite(LED_RED_1, HIGH); // гасим все красные полосковые индикаторы
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);
digitalWrite(LED_RED_CIRCUIT, HIGH);

digitalWrite(LED_BLUE_1, HIGH); // два раза мигаем синими светодиодами
digitalWrite(LED_BLUE_2, HIGH);
digitalWrite(LED_BLUE_3, HIGH);
digitalWrite(LED_BLUE_4, HIGH);
digitalWrite(LED_BLUE_5, HIGH);
delay (250);
digitalWrite(LED_BLUE_1, LOW);

```

```

digitalWrite(LED_BLUE_2, LOW);
digitalWrite(LED_BLUE_3, LOW);
digitalWrite(LED_BLUE_4, LOW);
digitalWrite(LED_BLUE_5, LOW);
delay (250);
digitalWrite(LED_BLUE_1, HIGH);
digitalWrite(LED_BLUE_2, HIGH);
digitalWrite(LED_BLUE_3, HIGH);
digitalWrite(LED_BLUE_4, HIGH);
digitalWrite(LED_BLUE_5, HIGH);
delay (250);
digitalWrite(LED_BLUE_1, LOW);
digitalWrite(LED_BLUE_2, LOW);
digitalWrite(LED_BLUE_3, LOW);
digitalWrite(LED_BLUE_4, LOW);
digitalWrite(LED_BLUE_5, LOW);

```

```

digitalWrite(LED_BLUE_1, LOW); // индикация номера режима синими светодиодами
digitalWrite(LED_BLUE_2, LOW);
digitalWrite(LED_BLUE_3, LOW);
digitalWrite(LED_BLUE_4, LOW);
digitalWrite(LED_BLUE_5, LOW);

```

```

switch (reg)
{
  case 1: digitalWrite(LED_BLUE_1, HIGH); break;
  case 2: digitalWrite(LED_BLUE_2, HIGH); break;
  case 3: digitalWrite(LED_BLUE_3, HIGH); break;
  case 4: digitalWrite(LED_BLUE_4, HIGH); break;
  case 5: digitalWrite(LED_BLUE_5, HIGH); break;
}

```

```

while(!red_button_pressed)
{
  while(!blue_button_pressed) // ожидание нажатия синей кнопки
  {
    if (digitalRead (BLUE_BUTTON) == HIGH) blue_button_pressed = true;

    if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true; // если нажата
    большая красная кнопка - прерывание цикла (режим выбран)
    if (red_button_pressed) break;
  }
  if (red_button_pressed) break; // если нажата большая красная кнопка - прерывание цикла
  (режим выбран)
}

```

```

digitalWrite(BEEPER, HIGH); // звуковой сигнал
delay(10);
digitalWrite(BEEPER, LOW);

```

```

reg++; // изменение номера режима
if(reg > 5) reg = 1;

```

```

digitalWrite(LED_BLUE_1, LOW); // индикация номера режима синими светодиодами
digitalWrite(LED_BLUE_2, LOW);
digitalWrite(LED_BLUE_3, LOW);
digitalWrite(LED_BLUE_4, LOW);
digitalWrite(LED_BLUE_5, LOW);
switch (reg)
{
  case 1: digitalWrite(LED_BLUE_1, HIGH); break;
  case 2: digitalWrite(LED_BLUE_2, HIGH); break;
  case 3: digitalWrite(LED_BLUE_3, HIGH); break;
  case 4: digitalWrite(LED_BLUE_4, HIGH); break;
  case 5: digitalWrite(LED_BLUE_5, HIGH); break;
}

delay(500); // подавлениедребезга синей кнопки

while(blue_button_pressed) // ожиданиеотжатия синей кнопки
{
  if (digitalRead (BLUE_BUTTON) == LOW) blue_button_pressed = false;
}

if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true; // если нажата
большая красная кнопка - прерывание цикла (режим выбран)
}

digitalWrite(BEEPER, HIGH); // звуковой сигнал - параметр выбран
delay(100);
digitalWrite(BEEPER, LOW);

delay(500);

while(red_button_pressed) // ожиданиеотжатия красной кнопки
if (digitalRead (BIG_RED_BUTTON) == LOW) red_button_pressed = false;

//*****
// работа вибродвигателя

blue_button_pressed = false;

switch (reg)
{
//*****
case 1: // постоянное включение на полную мощность
  work = false;

  digitalWrite(TRANSISTOR_1, LOW);
  digitalWrite(LED_BIG_BUTTON, LOW);

  while(!red_button_pressed) // ожидание нажатия красной кнопки
  {
    digitalWrite(LED_BIG_BUTTON, HIGH);

```

```

    delay (50);
    if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
    digitalWrite(LED_BIG_BUTTON, LOW);
    delay (50);
    if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
    if (red_button_pressed) break;
}

digitalWrite(BEEPER, HIGH); // звуковой сигнал - предупреждение о пуске
delay(1000);
digitalWrite(BEEPER, LOW);

// работа по case 1

while(!blue_button_pressed)
{
    if (work)
    {
        digitalWrite(TRANSISTOR_1, HIGH);
        digitalWrite(LED_BIG_BUTTON, HIGH);
    }
    else
    {
        digitalWrite(TRANSISTOR_1, LOW);
        digitalWrite(LED_BIG_BUTTON, LOW);
    }

    if (digitalRead (BIG_RED_BUTTON) == HIGH) // переключение work при нажатии
    большой красной кнопки
    {
        work = !work;
        delay (500); // подавление дребезга большой красной кнопки
    }

    if (digitalRead (BLUE_BUTTON) == HIGH) blue_button_pressed = true;
    if (blue_button_pressed) break;
}

// выход из case 1

digitalWrite(TRANSISTOR_1, LOW);
digitalWrite(LED_BIG_BUTTON, LOW);

digitalWrite(BEEPER, HIGH); // звуковой сигнал - завершение работы
delay(1000);
digitalWrite(BEEPER, LOW);

red_button_pressed = false;
break;

```

```

//*****
case 2: // изменение частоты при скважности 2

    digitalWrite(LED_RED_CIRCUIT, LOW); // два раза мигаем контуром полоскового
индикатора
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, HIGH);
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, LOW);
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, HIGH);

    while(!red_button_pressed)
    {
        digitalWrite(LED_RED_1, HIGH);
        digitalWrite(LED_RED_2, HIGH);
        digitalWrite(LED_RED_3, HIGH);
        digitalWrite(LED_RED_4, HIGH);
        digitalWrite(LED_RED_5, HIGH);
        digitalWrite(LED_RED_CIRCUIT, HIGH);

        Uvr = analogRead (VAR_RESISTOR);

        if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
        if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
        if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
        if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
        if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

        if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true; // если нажата
большая красная кнопка - прерывание цикла (параметр выбран)
    }

    digitalWrite(LED_RED_CIRCUIT, LOW);

    digitalWrite(BEEPER, HIGH); // звуковой сигнал - параметр выбран
    delay(100);
    digitalWrite(BEEPER, LOW);

    delay(500); // подавление дребезга большой красной кнопки

    while(red_button_pressed) // ожидание отжатия красной кнопки
        if (digitalRead (BIG_RED_BUTTON) == LOW) red_button_pressed = false;

    while(!red_button_pressed) // ожидание нажатия красной кнопки
    {
        digitalWrite(LED_BIG_BUTTON, HIGH);
        delay (50);
        if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
        digitalWrite(LED_BIG_BUTTON, LOW);
        delay (50);
        if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
    }

```

```

    if (red_button_pressed) break;
}

digitalWrite(BEEPER, HIGH); // звуковой сигнал - предупреждение о пуске
delay(1000);
digitalWrite(BEEPER, LOW);

// работа по case 2

work = false;
while(!blue_button_pressed)
{
    digitalWrite(LED_RED_1, HIGH);
    digitalWrite(LED_RED_2, HIGH);
    digitalWrite(LED_RED_3, HIGH);
    digitalWrite(LED_RED_4, HIGH);
    digitalWrite(LED_RED_5, HIGH);

    Uvr = analogRead (VAR_RESISTOR);

    if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
    if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
    if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
    if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
    if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

    T = 1000;
    if (Uvr > 0) T = 1000/(0.048876 * Uvr);
    if (T > 1000) T = 1000;

    if (digitalRead (BIG_RED_BUTTON) == HIGH)
    {
        time = millis ();
        if (time - time_pred > 500) // подавление дребезга большой красной кнопки
        {
            work = !work;
            time_pred = millis();
        }
    }

    if (work)
    {
        digitalWrite(TRANSISTOR_1, HIGH);
        digitalWrite(LED_BIG_BUTTON, HIGH);
    }

    delay(T/2);

    digitalWrite(TRANSISTOR_1, LOW);
    digitalWrite(LED_BIG_BUTTON, LOW);

```

```

delay(T/2);

if (digitalRead (BLUE_BUTTON) == HIGH) blue_button_pressed = true;
if (blue_button_pressed) break;
}

// ВЫХОД ИЗ case 2

digitalWrite(TRANSISTOR_1, LOW);
digitalWrite(LED_BIG_BUTTON, LOW);

digitalWrite(LED_BIG_BUTTON, LOW); // гасим большую красную кнопку
digitalWrite(LED_RED_1, HIGH); // гасим все красные полосковые индикаторы
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);
digitalWrite(LED_RED_CIRCUIT, HIGH);

digitalWrite(BEEPER, HIGH); // звуковой сигнал - завершение работы
delay(1000);
digitalWrite(BEEPER, LOW);

delay(500); // подавление дребезга синей кнопки

red_button_pressed = false;
break;

//*****
case 3: // изменение скважности при частоте 60 Гц

digitalWrite(LED_RED_CIRCUIT, LOW); // два раза мигаем контуром полоскового
индикатора
delay (250);
digitalWrite(LED_RED_CIRCUIT, HIGH);
delay (250);
digitalWrite(LED_RED_CIRCUIT, LOW);
delay (250);
digitalWrite(LED_RED_CIRCUIT, HIGH);

while(!red_button_pressed)
{
digitalWrite(LED_RED_1, HIGH);
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);
digitalWrite(LED_RED_CIRCUIT, HIGH);
}

```

```

Uvr = analogRead (VAR_RESISTOR);

if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true; // если нажата
большая красная кнопка - прерывание цикла (параметр выбран)
}

digitalWrite(LED_RED_CIRCUIT, LOW);

digitalWrite(BEEPER, HIGH); // звуковой сигнал - параметр выбран
delay(100);
digitalWrite(BEEPER, LOW);

delay(500); // подавление дребезга большой красной кнопки

while(red_button_pressed) // ожидание отжатия красной кнопки
if (digitalRead (BIG_RED_BUTTON) == LOW) red_button_pressed = false;

while(!red_button_pressed) // ожидание нажатия красной кнопки
{
digitalWrite(LED_BIG_BUTTON, HIGH);
delay (50);
if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
digitalWrite(LED_BIG_BUTTON, LOW);
delay (50);
if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
if (red_button_pressed) break;
}

digitalWrite(BEEPER, HIGH); // звуковой сигнал - предупреждение о пуске
delay(1000);
digitalWrite(BEEPER, LOW);

// работа по case 3

work = false;
T = 16,6667;

while(!blue_button_pressed)
{
digitalWrite(LED_RED_1, HIGH);
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);

Uvr = analogRead (VAR_RESISTOR);

```

```

if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

//S = 10*Uvr/1023;
S = -0.0098*Uvr + 10;
//S = 9/1023*Uvr + 1;
//S = 59.06/(Uvr + 1) + 0.942;
if (S < 1) S = 1;
tau = T/S;

if (digitalRead (BIG_RED_BUTTON) == HIGH)
{
time = millis ();
if (time - time_pred > 500)
{
work = !work;
time_pred = millis();
}
}

if (work)
{
digitalWrite(TRANSISTOR_1, HIGH);
digitalWrite(LED_BIG_BUTTON, HIGH);
}

delay(tau);

if (S > 1)
{
digitalWrite(TRANSISTOR_1, LOW);
digitalWrite(LED_BIG_BUTTON, LOW);
delay(T - tau);
}

if (digitalRead (BLUE_BUTTON) == HIGH) blue_button_pressed = true;
if (blue_button_pressed) break;
}

digitalWrite(TRANSISTOR_1, LOW);

digitalWrite(LED_BIG_BUTTON, LOW); // гасим большую красную кнопку
digitalWrite(LED_RED_1, HIGH); // гасим все красные полосковые индикаторы
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);

```

```

digitalWrite(LED_RED_5, HIGH);
digitalWrite(LED_RED_CIRCUIT, HIGH);

digitalWrite(BEEPER, HIGH); // звуковой сигнал - завершение работы
delay(1000);
digitalWrite(BEEPER, LOW);

red_button_pressed = false;
break;

//*****
case 4: // изменение скважности при заданной частоте

    digitalWrite(LED_RED_CIRCUIT, LOW); // два раза мигаем контуром полоскового
индикатора
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, HIGH);
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, LOW);
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, HIGH);

    while(!red_button_pressed)
    {
        digitalWrite(LED_RED_1, HIGH);
        digitalWrite(LED_RED_2, HIGH);
        digitalWrite(LED_RED_3, HIGH);
        digitalWrite(LED_RED_4, HIGH);
        digitalWrite(LED_RED_5, HIGH);
        digitalWrite(LED_RED_CIRCUIT, HIGH);

        Uvr = analogRead (VAR_RESISTOR);

        if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
        if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
        if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
        if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
        if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

        if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true; // если нажата
большая красная кнопка - прерывание цикла (частота выбрана)
    }

    digitalWrite(LED_RED_CIRCUIT, LOW);

    digitalWrite(BEEPER, HIGH); // звуковой сигнал - частота выбрана
    delay(100);
    digitalWrite(BEEPER, LOW);

    delay(500); // подавление дребезга большой красной кнопки

```

```

while(red_button_pressed) // ожидание отжатия красной кнопки
  if (digitalRead (BIG_RED_BUTTON) == LOW) red_button_pressed = false;

T = 1000;
if (Uvr > 0) T = 1000/(0.048876 * Uvr);
if (T > 1000) T = 1000; // частота выбрана

digitalWrite(LED_RED_CIRCUIT, LOW); // два раза мигаем контуром полоскового
индикатора
delay (250);
digitalWrite(LED_RED_CIRCUIT, HIGH);
delay (250);
digitalWrite(LED_RED_CIRCUIT, LOW);
delay (250);
digitalWrite(LED_RED_CIRCUIT, HIGH);

while(!red_button_pressed)
{
  digitalWrite(LED_RED_1, HIGH);
  digitalWrite(LED_RED_2, HIGH);
  digitalWrite(LED_RED_3, HIGH);
  digitalWrite(LED_RED_4, HIGH);
  digitalWrite(LED_RED_5, HIGH);

  Uvr = analogRead (VAR_RESISTOR);

  if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
  if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
  if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
  if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
  if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

  S = -0.0098*Uvr + 10;

  if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true; // если нажата
большая красная кнопка - прерывание цикла (скважность выбрана)
}

digitalWrite(LED_RED_CIRCUIT, LOW);
// выбрана скважность

delay (500); // подавление дребезга большой красной кнопки
red_button_pressed = false;
while(!red_button_pressed) // ожидание нажатия большой красной кнопки (с миганием)
{
  digitalWrite(LED_BIG_BUTTON, HIGH);
  delay (50);
  if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
  digitalWrite(LED_BIG_BUTTON, LOW);
  delay (50);
}

```

```

    if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
}

digitalWrite(BEEPER, HIGH); // звуковой сигнал - предупреждение о пуске
delay(1000);
digitalWrite(BEEPER, LOW);

// работа по case 4
work = false;
while(!blue_button_pressed)
{
    digitalWrite(LED_RED_1, HIGH);
    digitalWrite(LED_RED_2, HIGH);
    digitalWrite(LED_RED_3, HIGH);
    digitalWrite(LED_RED_4, HIGH);
    digitalWrite(LED_RED_5, HIGH);

    Uvr = analogRead (VAR_RESISTOR);

    if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
    if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
    if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
    if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
    if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

    S = -0.0098*Uvr + 10;
    if (S < 1) S = 1;

    tau = T/S;

    if (digitalRead (BIG_RED_BUTTON) == HIGH)
    {
        time = millis ();
        if (time - time_pred > 500)
        {
            work = !work;
            time_pred = millis();
        }
    }

    if (work)
    {
        digitalWrite(TRANSISTOR_1, HIGH);
        digitalWrite(LED_BIG_BUTTON, HIGH);
    }
    else
    {
        digitalWrite(TRANSISTOR_1, LOW);
    }
}

```

```

    digitalWrite(LED_BIG_BUTTON, LOW);
}

delay(tau);

if (S > 1)
{
    digitalWrite(TRANSISTOR_1, LOW);
    digitalWrite(LED_BIG_BUTTON, LOW);
    delay(T - tau);
}

if (digitalRead (BLUE_BUTTON) == HIGH) blue_button_pressed = true;
if (blue_button_pressed) break;
}

// ВЫХОД ИЗ case 4
digitalWrite(TRANSISTOR_1, LOW);
digitalWrite(LED_BIG_BUTTON, LOW);

digitalWrite(LED_RED_1, HIGH); // гасим все красные полосковые индикаторы
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);
digitalWrite(LED_RED_CIRCUIT, HIGH);

digitalWrite(BEEPER, HIGH); // звуковой сигнал - завершение работы
delay(1000);
digitalWrite(BEEPER, LOW);

delay(500); // подавление дребезга синей кнопки

red_button_pressed = false;
break;

//*****
case 5: // изменение частоты при заданной скважности

    digitalWrite(LED_RED_CIRCUIT, LOW); // два раза мигаем контуром полоскового
индикатора
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, HIGH);
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, LOW);
    delay (250);
    digitalWrite(LED_RED_CIRCUIT, HIGH);

while(!red_button_pressed)
{

```

```

digitalWrite(LED_RED_1, HIGH);
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);
digitalWrite(LED_RED_CIRCUIT, HIGH);

Uvr = analogRead (VAR_RESISTOR);

if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true; // если нажата
большая красная кнопка - прерывание цикла (частота выбрана)
}

digitalWrite(LED_RED_CIRCUIT, LOW);

digitalWrite(BEEPER, HIGH); // звуковой сигнал - частота выбрана
delay(100);
digitalWrite(BEEPER, LOW);

delay(500); // подавление дребезга большой красной кнопки

while(red_button_pressed) // ожидание отжатия красной кнопки
  if (digitalRead (BIG_RED_BUTTON) == LOW) red_button_pressed = false;

S = -0.0098*Uvr + 10;
if (S < 1) S = 1;

digitalWrite(LED_RED_CIRCUIT, LOW); // два раза мигаем контуром полоскового
индикатора
delay (250);
digitalWrite(LED_RED_CIRCUIT, HIGH);
delay (250);
digitalWrite(LED_RED_CIRCUIT, LOW);
delay (250);
digitalWrite(LED_RED_CIRCUIT, HIGH);

while(!red_button_pressed)
{
  digitalWrite(LED_RED_1, HIGH);
  digitalWrite(LED_RED_2, HIGH);
  digitalWrite(LED_RED_3, HIGH);
  digitalWrite(LED_RED_4, HIGH);
  digitalWrite(LED_RED_5, HIGH);

  Uvr = analogRead (VAR_RESISTOR);

```

```

if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);
if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

T = 1000;
if (Uvr > 0) T = 1000/(0.048876 * Uvr);
if (T > 1000) T = 1000; // частота выбрана

if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true; // если нажата
большая красная кнопка - прерывание цикла (частота выбрана)
}

digitalWrite(LED_RED_CIRCUIT, LOW);
// выбрана частота

delay (500); // подавление дребезга большой красной кнопки
red_button_pressed = false;
while(!red_button_pressed) // ожидание нажатия большой красной кнопки (с миганием)
{
digitalWrite(LED_BIG_BUTTON, HIGH);
delay (50);
if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
digitalWrite(LED_BIG_BUTTON, LOW);
delay (50);

if (digitalRead (BIG_RED_BUTTON) == HIGH) red_button_pressed = true;
}

digitalWrite(BEEPER, HIGH); // звуковой сигнал - предупреждение о пуске
delay(1000);
digitalWrite(BEEPER, LOW);

// работа по case 5
work = false;
while(!blue_button_pressed)
{
digitalWrite(LED_RED_1, HIGH);
digitalWrite(LED_RED_2, HIGH);
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);

Uvr = analogRead (VAR_RESISTOR);

if (Uvr > 1023/5) digitalWrite(LED_RED_1, LOW);
if (Uvr > 2 * 1023/5) digitalWrite(LED_RED_2, LOW);
if (Uvr > 3 * 1023/5) digitalWrite(LED_RED_3, LOW);

```

```

if (Uvr > 4 * 1023/5) digitalWrite(LED_RED_4, LOW);
if (Uvr >= 5 * 1023/5) digitalWrite(LED_RED_5, LOW);

T = 1000;
if (Uvr > 0) T = 1000/(0.048876 * Uvr);
if (T > 1000) T = 1000;

tau = T/S;

if (digitalRead (BIG_RED_BUTTON) == HIGH)
{
time = millis ();
if (time - time_pred > 500)
{
work = !work;
time_pred = millis();
}
}

if (work)
{
digitalWrite(TRANSISTOR_1, HIGH);
digitalWrite(LED_BIG_BUTTON, HIGH);
}
else
{
digitalWrite(TRANSISTOR_1, LOW);
digitalWrite(LED_BIG_BUTTON, LOW);
}

delay(tau);

if (S > 1)
{
digitalWrite(TRANSISTOR_1, LOW);
digitalWrite(LED_BIG_BUTTON, LOW);
delay(T - tau);
}

if (digitalRead (BLUE_BUTTON) == HIGH) blue_button_pressed = true;
if (blue_button_pressed) break;
}

// ВЫХОД ИЗ case 5
digitalWrite(TRANSISTOR_1, LOW);
digitalWrite(LED_BIG_BUTTON, LOW);

digitalWrite(LED_RED_1, HIGH); // гасим все красные полосковые индикаторы
digitalWrite(LED_RED_2, HIGH);

```

```
digitalWrite(LED_RED_3, HIGH);
digitalWrite(LED_RED_4, HIGH);
digitalWrite(LED_RED_5, HIGH);
digitalWrite(LED_RED_CIRCUIT, HIGH);

digitalWrite(BEEPER, HIGH); // звуковой сигнал - завершение работы
delay(1000);
digitalWrite(BEEPER, LOW);

delay(500); // подавление дребезга синей кнопки

red_button_pressed = false;
break;

}

//*****

}
```