МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных *наук*

Кафедра информационных систем

Цифровой измеритель фазы переменного тока на микропроцессоре

Отчет по учебной практике,
научно-исследовательской работе
09.03.02 Информационные системы и технологии
Информационные системы и сетевые технологии

6 семестр 2021/2022 учебного года

Зав. кафедрой _____ к. т. н., доцент Д.Н. Борисов

Обучающийся _____ ст. 3 курса оч. отд. С.А. Лемза

Руководитель _____ к. т. н., доцент С.А. Зуев

Воронеж 2022

Оглавление

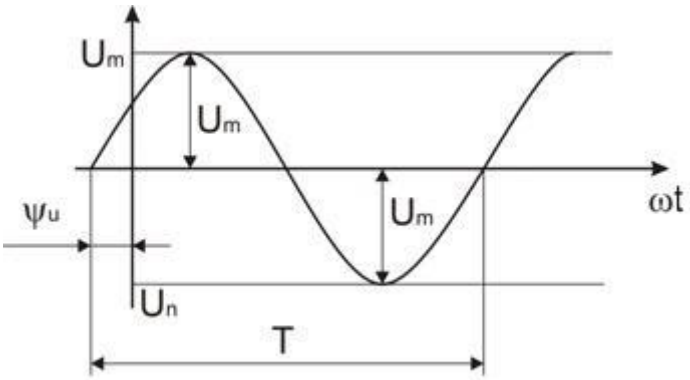
Введение....2
Используемые компоненты....3
Схема подключения....5
Код программы....7
Работа устройства....11
Активная, реактивная и полная мощность....11

Введение

Фазовый сдвиг – разность между начальными фазами двух переменных величин, изменяющихся во времени периодически с одинаковой частотой. Сдвиг фаз является величиной безразмерной и может измеряться в радианах (градусах) или долях периода. При неизменном, в частности нулевом сдвиге фаз говорят о синхронности двух процессов, или о выполненной синхронизации двух источников переменных величин.

Фазой (фазовым углом) называется угол φ = 2π*t/T, где T - период, t - доля периода смещения по фазе при наложении синусоид друг на друга.

Чтобы лучше понять значения термина фаза, обратимся к графику зависимости напряжения в однофазной сети переменного тока от времени. Здесь видно, что, напряжение изменяется от некоторого максимального значения Um до -Um, периодически проходя чрез ноль.



$$u(t) = U_m \sin(\omega t + \psi_u)$$

Рисунок 1

В процессе изменения, напряжение принимает множество значений в каждый момент времени, периодически (спустя период времени T) возвращаясь к тому значению, с которого начиналось наблюдение за данным напряжением. Можно сказать, что в любой момент времени напряжение находится в определенной фазе, которая зависит от нескольких факторов: от времени t, прошедшего от начала колебаний, от угловой частоты, и от начальной фазы. То, что стоит в скобках — полная фаза колебаний в текущий момент времени t. Пси — начальная фаза.

Начальную фазу называют в электротехнике еще начальным фазовым углом, поскольку фаза измеряется в радианах или в градусах, как и все обычные геометрические углы. Пределы изменения фазы лежат в интервале от 0 до 360 градусов или от 0 до 2*пи радиан.
На приведенном выше рисунке видно, что в момент начала наблюдения за переменным напряжением U, его значение не было нулем, то есть фаза уже успела в данном примере отклониться от нуля на некоторый угол Пси, равный около 30 градусов или пи/6 радиан — это и есть начальный фазовый угол.
При наличии в цепи переменного тока конденсатора или катушки индуктивности частота изменения тока и напряжения идентичны, но различны, если смотреть на графики, их начальные фазы. В этом случае говорят о фазовом сдвиге между током и напряжением, то есть о разности их начальных фазовых углов.

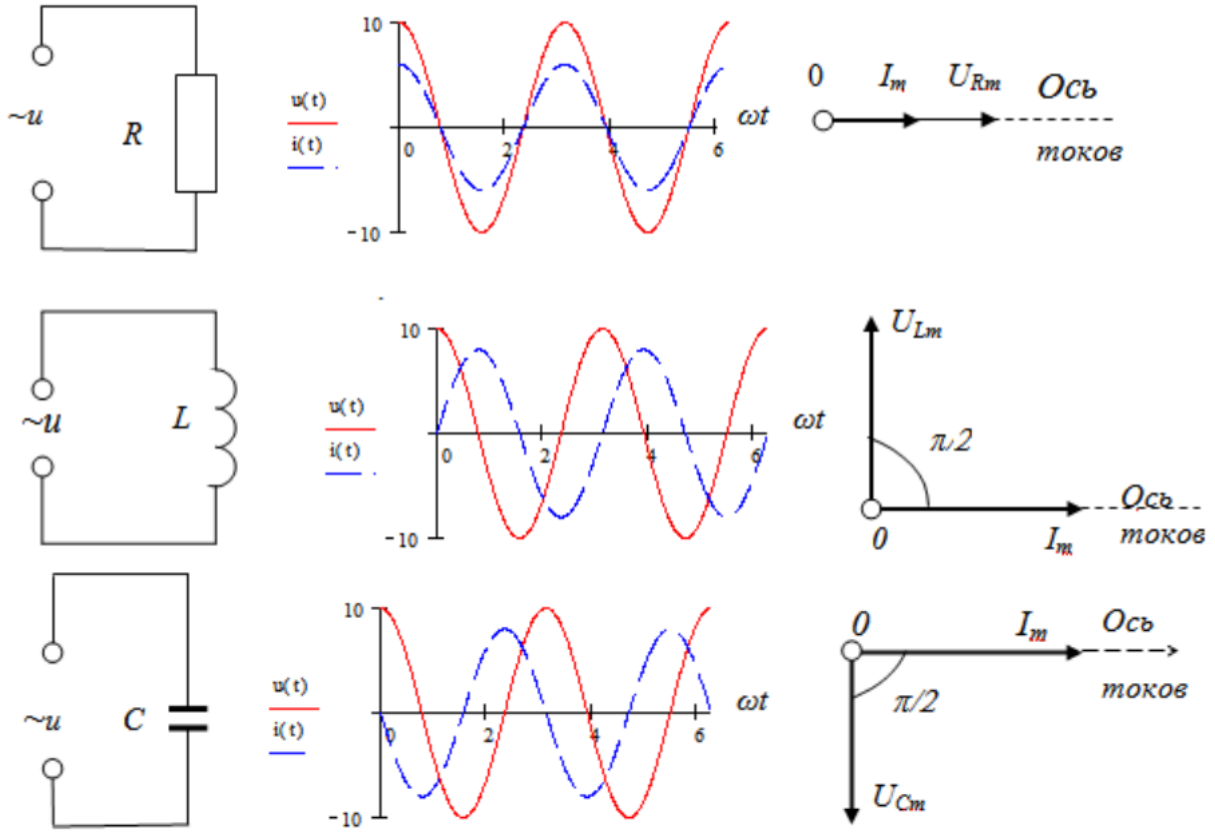


Рисунок 2

Используемые компоненты

STM32F103 имеет возможность измерять переменное напряжение или переменный ток с помощью аналогового входа. У неё есть 14 аналоговых входных контактов (A0-A15). Аналоговые входные контакты преобразуют входные напряжения от 0 до 5 В в целые числа от 0 до 1023 с разрешением 4,9 мВ на единицу (5,00 В / 1024 единицы). Для измерения фазового угла требуются 2 контакта аналогового входа.

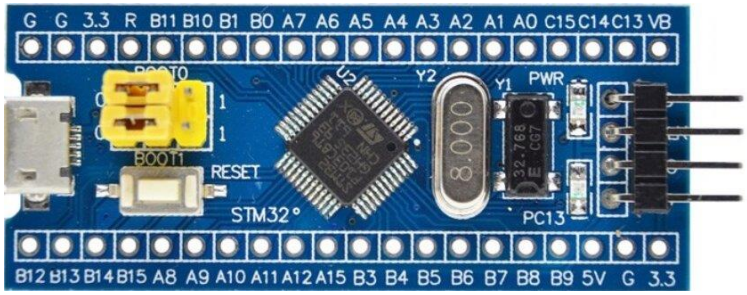


Рисунок 3

ZMPT101B - однофазный модуль напряжения переменного тока. Этот модуль оснащен высокоточным трансформатором напряжения ZMPT101B и схемой операционного усилителя. Он может измерять напряжение переменного тока в пределах 250 В. Соответствующий выходной сигнал можно отрегулировать с помощью подстроечного потенциометра.

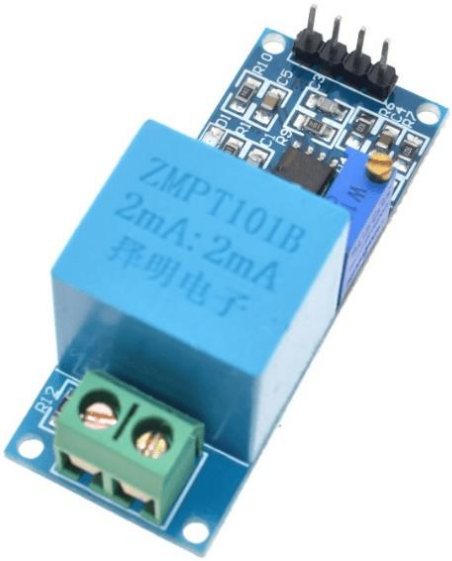


Рисунок 4

HSTS016L - датчика переменного тока. Этот датчик с разъемным сердечником на эффекте Холла может измерять переменный ток.

Рисунок 5

Схема подключения

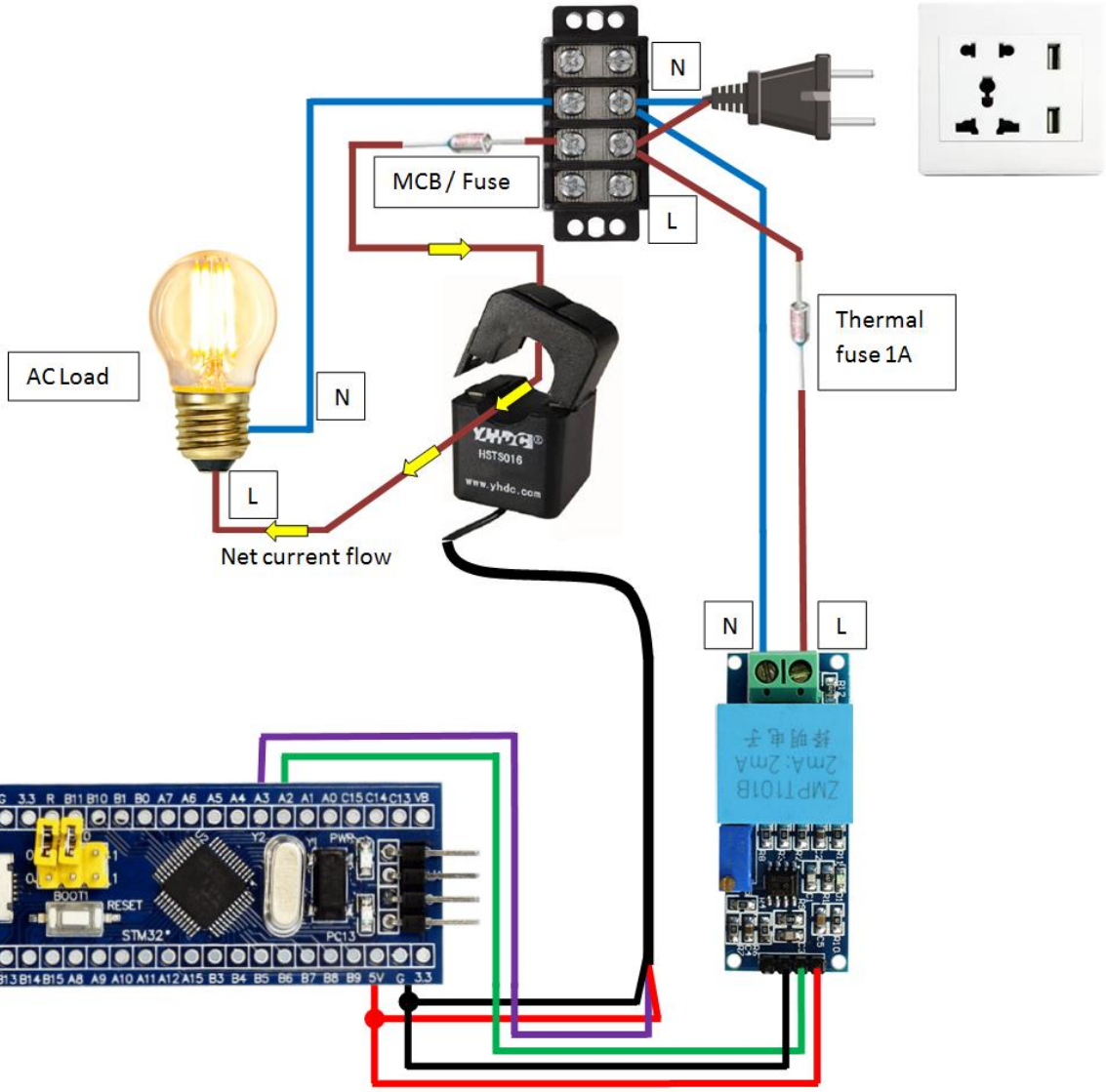
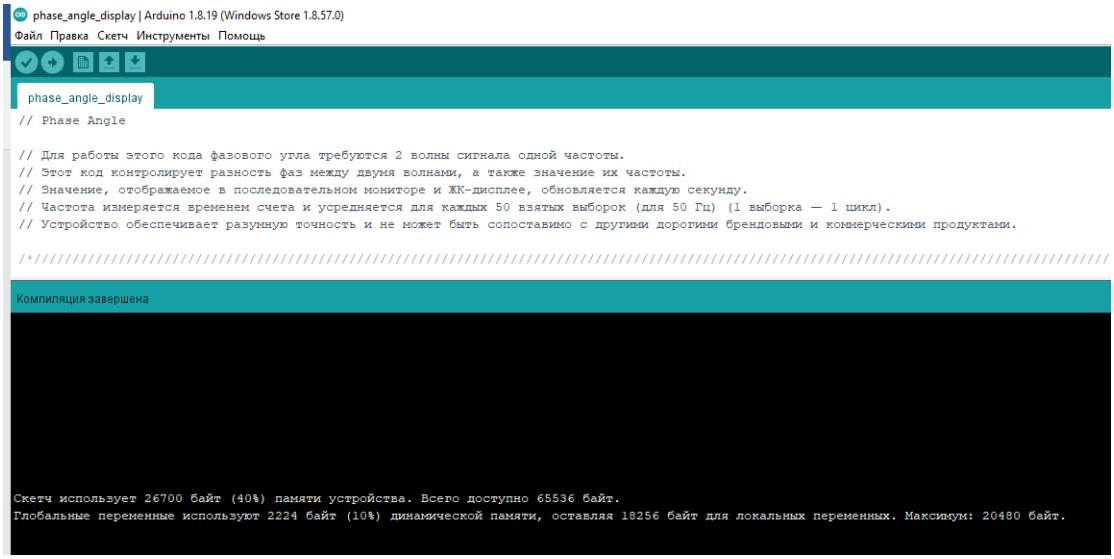


Рисунок 6

Скомпилируем и загрузим в STM32F103 программный код.



Сигналы, обнаруженные STM32F103, имеют аналоговые значения. Ниже приведены сигналы, обнаруженные датчиком напряжения (синий) и датчиком тока (красный). Можно увидеть форму сигнала в Serial Plotter.

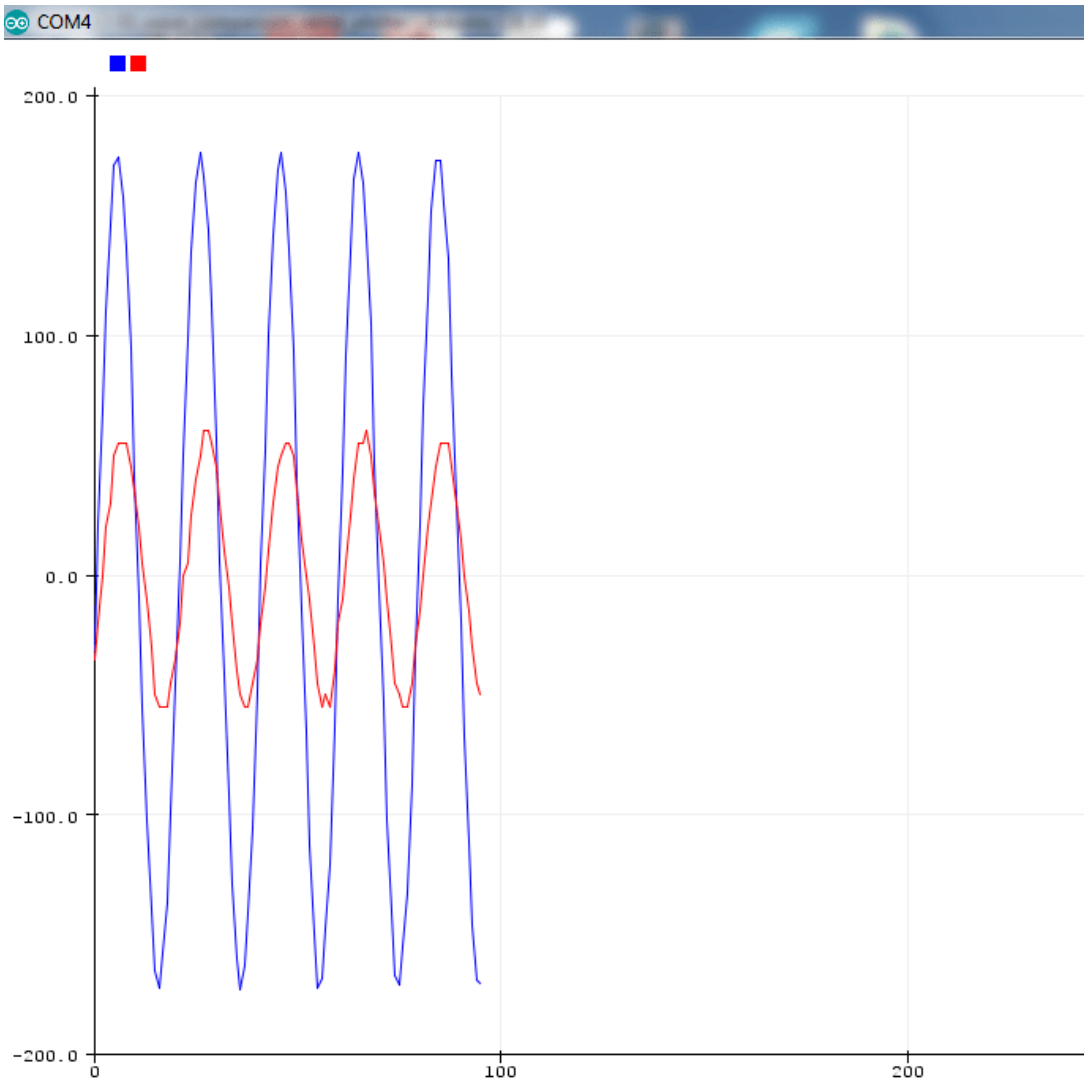


Рисунок 7

Для правильной работы устройства, необходимо произвести начальную калибровку обоих датчиков. Нужно свести к минимуму любую потенциальную ошибку отклонения или неточность, насколько это возможно. Датчики очень чувствительны, поэтому обязательно использование плотных разъемов и кабельных наконечников. При проведении калибровки, когда значение не обнаружено, оба датчика должны выдавать 0.

Код программы

```
// Phase Angle

// Для работы этого кода фазового угла требуются 2 волны сигнала одной частоты.
// Этот код контролирует разность фаз между двумя волнами, а также значение их частоты.
// Значение, отображаемое в последовательном мониторе и ЖК-дисплее, обновляется каждую секунду.
// Частота измеряется временем счета и усредняется для каждых 50 взятых выборок (для 50 Гц) (1 выборка — 1 цикл).
// Устройство обеспечивает разумную точность и не может быть сопоставимо с другими дорогими брендовыми и коммерческими продуктами.

/*////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*////////////*/

        /* 0- General */

        int decimalPrecision = 2;// десятичные разряды для всех значений, отображаемых на дисплее и в последовательном мониторе

        /* 1 - Измерение фазового угла, частоты и коэффициента мощности */

        int expectedFrequency = 50;// Введите частоту для основной сети (50/60 Гц)
        int analogInputPin1PA = PA1;// Входной контакт для датчика AnalogRead1.  Используйте значение датчика напряжения в качестве первого значения.
        int analogInputPin2PA = PA2;// Входной контакт для датчика AnalogRead2.  Используйте значение датчика тока в качестве второго значения.
        float voltageAnalogOffset =0;// Это для смещения аналогового значения для AnalogInput1
        float currentAnalogOffset =0;// Это для смещения аналогового значения для AnalogInput2
        unsigned long startMicrosPA;/* начать отсчет времени для фазового угла и периода (в микросекундах) */
        unsigned long vCurrentMicrosPA;/* текущее время для analogInput1 (напряжение) (в микросекундах). AnalogInput1 используется в качестве опорного для фазового угла*/
        unsigned long iCurrentMicrosPA;/* текущее время для analogInput2 (ток/напряжение) (в микросекундах).*/
        unsigned long periodMicrosPA;/* текущее время для периода записи волны */
        float vAnalogValue =0;/* это аналоговое значение для датчика напряжения AnalogInput1 */
        float iAnalogValue =0;/* аналоговое значение для датчика тока AnalogInput2 */
        float previousValueV =0;/* используется для записи пикового значения датчика напряжения */
        float previousValueI =0;/* используется для записи пикового значения датчика тока */
        float previousphaseAngleSample=0;/* предыдущее значение для замены ложного значения менее 100 микросекунд */
        float phaseAngleSample =0;/* разница во времени между показаниями двух датчиков (в микросекундах) */
        float phaseAngleAccumulate =0;/* разница во времени для накопления выборок */
        float periodSample=0;/* разница во времени для периода волны для выборки (в микросекундах) */
        float periodSampleAccumulate = 0;/* разница во времени для накопления выборок */
        float phaseDifference =0;/* это усредненный набор временной разницы 2-х датчиков */
        float phaseAngle =0; /* фазовый угол в градусах (из 360) */
        float frequency = 0;/* частота волны датчика напряжения */
        float voltagePhaseAngle=0;/* это время, записанное с момента начала достижения пикового значения для analogInput1 в микросекундах*/
        float currentPhaseAngle=0;/* это время, записанное с момента начала достижения пикового значения для analogInput2 в микросекундах*/
        float averagePeriod =0; /* средний набор времени, зарегистрированный за период волны */
        int sampleCount = 0; /* для подсчета количества наборов образцов */
        int a = 3; /* используем для операции переключения */
        float powerFactor;/* для расчета коэффициента мощности */

            /* 1.1 - Смещение фазового угла*/

            float currentOffsetRead =0;
            float currentOffsetLastSample =0;
            float currentOffsetSampleCount=0;
            float voltageOffsetRead =0;
```

```
        float voltageOffsetLastSample =0;
        float voltageOffsetSampleCount=0;


    /* 2 - ЖК-дисплей  */

    #include<LiquidCrystal.h>/* Загрузить библиотеку Crystal Library */
    LiquidCrystal LCD(8,9,4,5,6,7); /* Создание объекта LiquidCrystal */

void setup()
{
    /* 0- General */

    Serial.begin(9600);

    /* 2 - ЖК-дисплей  */

    LCD.begin(16,2); /* Сообщаем, что наш ЖК-дисплей имеет 16 столбцов и 2 строки*/
    LCD.setCursor(0,0); /* Установите курсора так, чтобы он начинался с верхнего левого угла дисплея*/
}

void loop()
{

    /* 0- General */

        /* 0.1- Button Function */

        int buttonRead;
        buttonRead = analogRead (0);
// Чтение аналогового вывода A0.  Контакт A0 автоматически назначается для функции кнопки ЖК-дисплея
        /* Нажата правая кнопка */
        if (buttonRead < 60)
        {   LCD.setCursor(0,0); LCD.print ("PRESS <SELECT>   "); }

        /* Нажата кнопка вверх */
        else if (buttonRead < 200)
        {   LCD.setCursor(0,0); LCD.print ("PRESS <SELECT>   "); }

        /* Нажата кнопка вниз */
        else if (buttonRead < 400)
        {   LCD.setCursor(0,0); LCD.print ("PRESS <SELECT>   "); }

        /* Нажата левая кнопка */
        else if (buttonRead < 600)
        {   LCD.setCursor(0,0); LCD.print ("PRESS <SELECT>   "); }

        /* Нажата кнопка выбора */
        else if (buttonRead < 800)
        {
        currentOffsetRead = 1; // чтобы активировать смещение для тока
        voltageOffsetRead = 1; // чтобы активировать смещение для напряжения
        LCD.setCursor(0,0); /* установить отображаемые слова, начиная с левого верхнего угла */
        LCD.print ("INITIALIZING..... ");
        LCD.setCursor(0,1); /* установить отображаемые слова, начиная с левого нижнего угла */
        LCD.print ("WAIT 5 SEC ..... ");
        }

    /* 1 - Измерение фазового угла, частоты и коэффициента мощности */

    vAnalogValue = analogRead(analogInputPin1PA)-512 + voltageAnalogOffset;
/* считывание analogInput1 с коррекцией по центру */
    iAnalogValue = analogRead(analogInputPin2PA)-512 + currentAnalogOffset;
/* считывание analogInput2 с коррекцией по центру */
```

```
      if((vAnalogValue>0) && a == 3)
/* начальный начальный этап измерения, когда волна AnalogInput1 больше 0 */
      {
       a=0; /* разрешить переход к следующему этапу */
      }
      if((vAnalogValue<=0) && a ==0)/* когда аналоговое значение AnalogInput1 меньше или равно 0 */
      {
       startMicrosPA = micros();/* начинаем считать время для всех */
       a=1;/* разрешить переход к следующему этапу */
      }
      if((vAnalogValue>0) && a ==1)/* когда аналоговое значение AnalogInput1 больше 0 */
      {
       a = 2;/* разрешить переход к следующему этапу */
       previousValueV = 0;
/* сбросить значение.  Это значение будет сравниваться с измерением пикового значения для analogInput1 */
       previousValueI = 0;
/* сбросить значение.  Это значение будет сравниваться с измерением пикового значения для analogInput2 */
      }

      if((vAnalogValue > previousValueV ) && a==2)
/* если текущее измеренное значение больше, чем предыдущее пиковое значение analogInput1 */
      {
       previousValueV = vAnalogValue ;
/* запись текущего значения измерения заменяет предыдущее пиковое значение */
       vCurrentMicrosPA = micros();
/* запись текущего времени для AnalogInput1 */
      }
      if((iAnalogValue > previousValueI) && a==2)
/* если текущее измеренное значение больше, чем предыдущее пиковое значение analogInput2 */
      {
       previousValueI = iAnalogValue ;
/* запись текущего значения измерения заменяет предыдущее пиковое значение */
       iCurrentMicrosPA = micros();/* запись текущего времени для AnalogInput2 */
      }
      if((vAnalogValue <=0) && a==2)
/* когда аналоговое значение AnalogInput1 меньше или равно 0 */
      {
       periodMicrosPA = micros();/* запись текущего времени за 1 период */
       periodSample = periodMicrosPA - startMicrosPA;
/* волна периода - это текущее время минус время начала (для 1 выборки) */
       periodSampleAccumulate = periodSampleAccumulate + periodSample;
/* суммируем время для всех показаний периодической волны */
       voltagePhaseAngle = vCurrentMicrosPA - startMicrosPA;
/* время, необходимое для analogInput1 от 0 (нисходящая волна) до пикового значения (восходящая волна)*/
       currentPhaseAngle = iCurrentMicrosPA - startMicrosPA;
/* время, необходимое для analogInput2 от 0 (нисходящая волна) до пикового значения (восходящая волна)*/
       phaseAngleSample = currentPhaseAngle - voltagePhaseAngle;
 /* разница во времени между пиковым значением analogInput1 и пиковым значением analogInput2 */
       if(phaseAngleSample>=100)
/* если разница во времени больше 100 микросекунд */
       {
       previousphaseAngleSample = phaseAngleSample;
/* заменить предыдущее значение новым текущим значением */
       }
       if(phaseAngleSample<100
/* если разница во времени меньше 100 микросекунд (может быть шум или поддельные значения) */
       {
       phaseAngleSample = previousphaseAngleSample;
 /* взять предыдущее значение вместо использования низкого */
       }
       phaseAngleAccumulate = phaseAngleAccumulate + phaseAngleSample;
 /* накапливаем или суммируем время для всех показаний разницы во времени */
       sampleCount = sampleCount + 1; /* подсчитываем номер выборки */
```

```
        startMicrosPA = periodMicrosPA /* сбросить время начала */
        a=1; /* сбросить режим стадии */
        previousValueV = 0; /* сброс пикового значения для analogInput1 для следующего набора */
        previousValueI = 0; /* сброс пикового значения для analogInput2 для следующего набора */
       }

      if(sampleCount == expectedFrequency)
/* если общее количество записанных семплов равно 50 по умолчанию */
       {
        averagePeriod = periodSampleAccumulate/sampleCount;
/* среднее время за период волны по всем показаниям выборки */
        frequency = 1000000 / averagePeriod; /* расчетное значение частоты */
        phaseDifference = phaseAngleAccumulate / sampleCount;
/* средняя разница во времени между пиковыми значениями двух датчиков из всех показаний выборки */
        phaseAngle = ((phaseDifference*360) / averagePeriod); /* расчетный фазовый угол в градусах (из 360) */
        powerFactor = cos(phaseAngle*0.017453292); /* коэффициента мощности */
        Serial.print("Phase Angle :");
        Serial.print(phaseAngle,decimalPrecision);
        Serial.print("Â°  ");
        Serial.print("Frequency :");
        Serial.print(frequency,decimalPrecision);
        Serial.print("Hz  ");
        Serial.print("Power Factor :");
        Serial.println(powerFactor,decimalPrecision);
        sampleCount = 0; /* сбросить количество подсчетов */
        periodSampleAccumulate = 0; /* сбрасываем накопленное время волны периода из всех выборок */
        phaseAngleAccumulate =0; /* сбрасываем накопленное время для разницы во времени всех выборок */

        /* 2 - LCD Display  */

        LCD.setCursor(0,0); /* Установить курсор на первый столбец 0 и вторую строку 1 */
        LCD.print("P. Angle =");
        LCD.print(phaseAngle,decimalPrecision);
/* отображаем текущее значение на ЖК-дисплее в первой строке */
        LCD.print((char)223);
        LCD.print("GR  ");
        LCD.setCursor(0,1);
        LCD.print(frequency,decimalPrecision);
        LCD.print("Hz  ");
        LCD.print(powerFactor,decimalPrecision);
/* отображаем текущее значение на ЖК-дисплее в первой строке */
        LCD.print("PF   ");

       }

         /* 1.1 - Смещение фазового угла */

         if( voltageOffsetRead == 1)
          {
           voltageAnalogOffset = 0;
           if(millis()>= voltageOffsetLastSample + 1) /* сохранить время подсчета для offset1 */
              {
                voltageOffsetSampleCount = voltageOffsetSampleCount + 1;
/* 1 миллисекунда добавляет 1 счет */
                voltageOffsetLastSample = millis();
/* чтобы снова сбросить время, чтобы следующий цикл мог начаться снова */
              }
            if(voltageOffsetSampleCount == 1500)
              {
                vAnalogValue = analogRead(analogInputPin1PA)-512 + voltageAnalogOffset;
/* считывание analogInput1 с коррекцией по центру*/
                voltageAnalogOffset = -1*(vAnalogValue); /* устанавливаем значения смещения */
                voltageOffsetRead = 0; /* перейти к настройкам второго смещения */
                voltageOffsetSampleCount = 0;
```

```
 /* чтобы снова сбросить время, чтобы следующий цикл мог начаться снова */
                }
            }
          if( currentOffsetRead == 1)
            {
              currentAnalogOffset = 0;
              if(millis()>= currentOffsetLastSample + 1)
/* сохранить время подсчета для offset1 */
                {
                  currentOffsetSampleCount = currentOffsetSampleCount + 1; /* 1 миллисекунда добавляет 1 счет */
                  currentOffsetLastSample = millis();
 /* чтобы снова сбросить время, чтобы следующий цикл мог начаться снова */
                }
               if(currentOffsetSampleCount == 1500)
                {
                  iAnalogValue = analogRead(analogInputPin2PA)-512 + currentAnalogOffset;
 /* считывание analogInput2 с регулировкой по центру*/
                  currentAnalogOffset = -1*(iAnalogValue); /* устанавливаем значения смещения */
                  currentOffsetRead = 0; /* перейти к настройкам второго смещения */
                  currentOffsetSampleCount = 0;
 /* чтобы снова сбросить время, чтобы следующий цикл мог начаться снова */
                }
            }
}
```

Работа устройства

Рисунок 8

Активная, реактивная и полная мощность

Нагрузка электрической цепи определяет, какой ток через неё проходит. Если ток постоянный, то эквивалентом нагрузки в большинстве случаев можно определить резистор определённого сопротивления. Тогда мощность рассчитывают по одной из формул: **$P=U*I$** **$P=I^2*R$** **$P=U^2/R$**

По этой же формуле определяется полная мощность в цепи переменного тока.

Нагрузку разделяют на два основных типа:

- Активную – это резистивная нагрузка, типа – ТЭНов, ламп накаливания и подобного.

- Реактивную – она бывает индуктивной (двигатели, катушки пускателей, соленоиды) и емкостной (конденсаторные установки и прочее).

Рисунок 9

Буквой P – обозначена активная мощность, Q – реактивная, S – полная.
Формула полной мощности имеет вид:

$$S = \sqrt{P^2 + Q^2}$$

Потребление активной составляющей электроэнергии: **P=S*cosФ**
Коэффициент мощности Ф – это угол между активной и полной составляющей из треугольника. Тогда: **cosФ=P/S**
В свою очередь реактивная мощность рассчитывается по формуле:
Q = U*I*sinФ
Какую работу выполняет реактивная мощность? Полезной работы она не выполняет, но нагрузкой на линию является полная мощность, в том числе с учетом реактивной составляющей. Поэтому чтобы снизить общую нагрузку с ней борются или говоря грамотным языком компенсируют.
Как её компенсируют? В этих целях используют установки для компенсации реактивной мощности. Это могут быть конденсаторные установки или синхронные компенсаторы.
Из-за каких потребителей возникает реактив? Это в первую очередь электродвигатели – самый многочисленный вид электрооборудования на предприятиях.
Чем вредит большое потребление реактивной энергии? Кроме нагрузки на линии электропередач следует учитывать, что предприятия оплачивает полную мощность. Это приводит к повышенной сумме оплаты за электроэнергию.