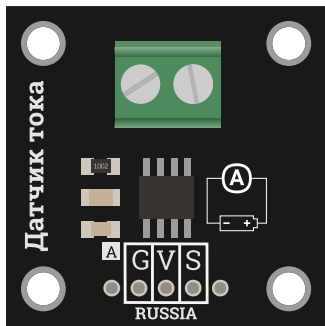


Датчик тока (Трета-модуль V2.0)



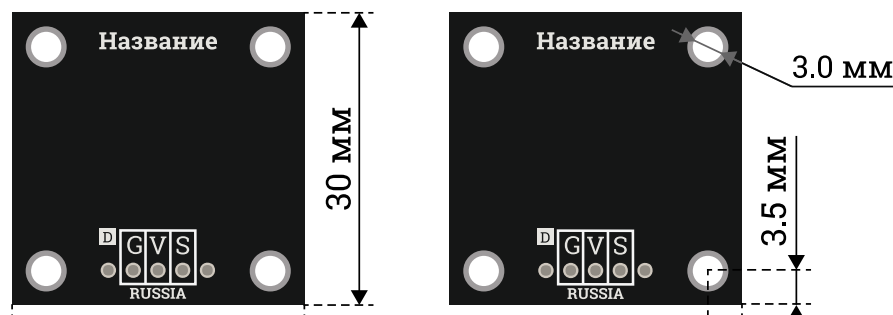
Общие сведения:

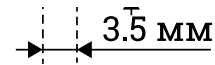
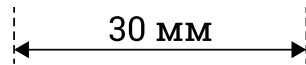
[Трета-модуль Датчик тока](#) - это аналоговый модуль, позволяющий определять силу как постоянного, так и переменного тока до 5А. Для определения силы тока протекающего по исследуемой цепи, нужно один из её проводов подключить через клеммник на плате модуля. Потенциал на выходе модуля «S» (Signal) будет меняться в соответствии с направлением и силой измеряемого тока.

Спецификация:

- Чип ACS712ELCTR-05B-T.
- Питание модуля: 5 В постоянного тока.
- Измеряемый постоянный ток: ± 5 А.
- Измеряемый переменный ток: ~ 5 А.
- Максимальное напряжение исследуемой цепи: 300 В.
- Частота измеряемого переменного тока: до 60 Гц.
- Сопротивление между выводами клеммника: 1,2 мОм.
- Потенциал на выходе «S» при отсутствии измеряемого тока: $V_{cc}/2$ В.
- Чувствительность модуля: 130,7 мВ/А. (Ток в 1 А смещает потенциал выхода «S» на 130,7 мВ).
- Рабочая температура: -40 ... 85 °С.
- Габариты модуля 30x30 мм.

Все модули линейки "Тема" выполнены в одном формате





Подключение:

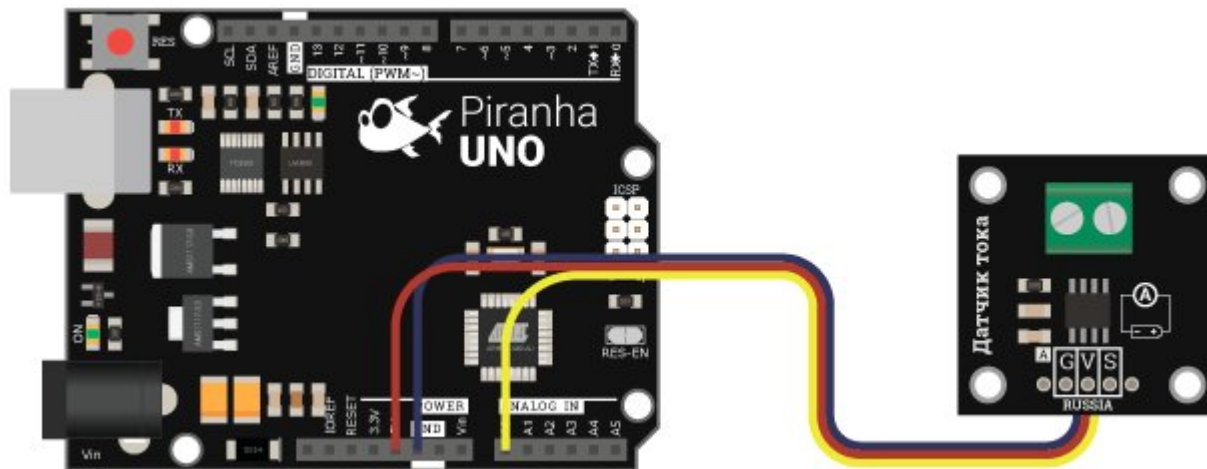
[Trema-модуль Датчик тока \(5A\)](#) является аналоговым модулем, а значит его выход «S» (Signal) подключается к любому аналоговому входу [Arduino](#).

В комплекте имеется кабель для быстрого и удобного подключения модуля к [Trema Shield](#).

Модуль удобно подключать 3 способами, в зависимости от ситуации:

Способ - 1 : Используя проводной шлейф и Piranha UNO

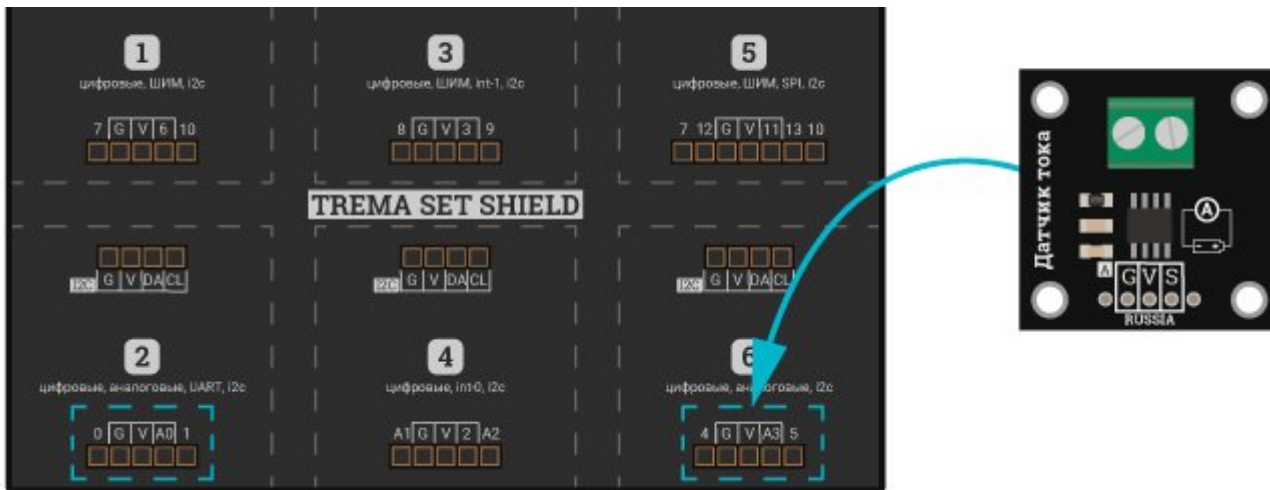
Используя провода «Папа – Мама», подключаем напрямую к контроллеру Piranha UNO



Способ - 2 : Используя Trema Set Shield

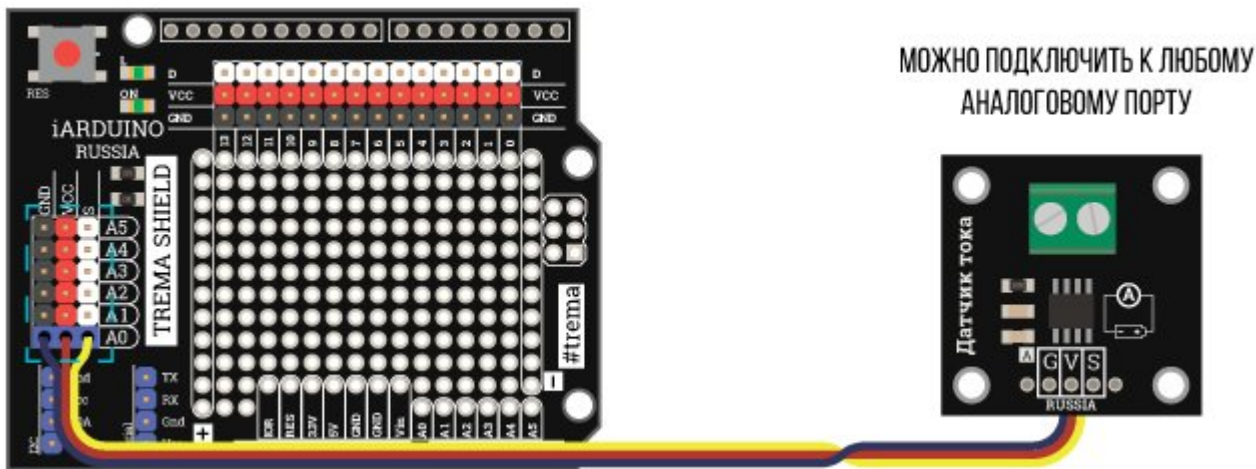
Модуль можно подключить к любому из аналоговых входов Trema Set Shield.





Способ - 3 : Используя проводной шлейф и Shield

Используя 3-х проводной шлейф, к Trema Shield, Trema-Power Shield, Motor Shield, Trema Shield NANO и тд.



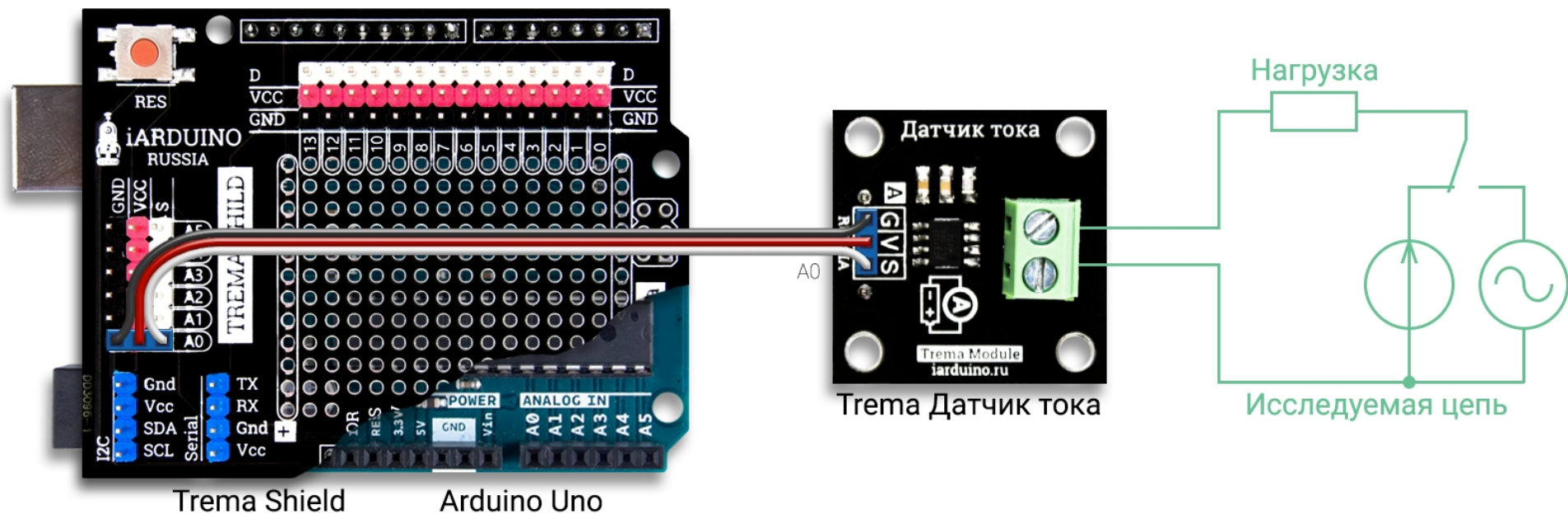
Питание:

Входное напряжение питания 5 В постоянного тока, подаётся на выводы «V» (Vcc) и «G» (GND) модуля.

Подробнее о модуле:

[Трема-модуль Датчик тока \(5A\)](#) построен на базе чипа ACS712 (Analog Current Sensor). Выводы чипа подключённые к клеммнику модуля соединены внутри чипа медной дорожкой, расположенной вблизи датчика Холла. При наличии тока протекающего по этой медной дорожке, генерируемое им магнитное поле, воспринимается датчиком Холла и преобразуется в напряжение на выходе чипа, пропорциональное силе тока. Таким образом выводы подключаемые к исследуемой цепи (ток которой измеряется) электрически изолированы от выхода и шины питания модуля. Для улучшения соотношения сигнал-шум, на выходе Трема модуля установлен RC-фильтр.

Для определения силы тока протекающего по исследуемой цепи, нужно один из проводов этой цепи подключить через клеммник на плате модуля.



Ток протекающий через клеммы [Трема-модуля Датчик тока \(5A\)](#) генерирует магнитное поле, которое воспринимается датчиком Холла и преобразуется в напряжение на выходе «S» модуля, пропорциональное силе и направлению измеряемого тока.

Для работы с модулем предлагаем воспользоваться библиотекой [iarduino_ACS712](#), в которой реализовано по 4 функции для работы с постоянным и переменным током:

- 4 функции для работы с постоянным током: `setZeroVDC()`, `getZeroVDC()`, `readDC()`, `calibrationDC()`.
- 4 функции для работы с переменным током: `setZeroVAC()`, `getZeroVAC()`, `readAC()`, `calibrationAC()`.

Перечисленные функции идентичны по назначению и отличаются лишь окончанием DC (для постоянного тока) / AC (для переменного тока).

Чтение силы тока осуществляется вызовом функции `readDC()` или `readAC()`, например, `float i = readDC()` // прочитать силу постоянного тока в переменную `i`. Функция не только возвращает силу тока в амперах, но и автоматически сглаживает выводимые показания (чем чаще вызывается функция, тем сильнее сглаживание показаний - показания меняются плавнее).

Если показания возвращаемые функцией `readDC()` или `readAC()` отличаются от реально измеренных, то их можно откорректировать вызвав функцию `calibrationDC()` или `calibrationAC()`, с параметром в виде числа от 0 до 1023 (по умолчанию 511). Данная функция удобна если для корректировки показаний датчика тока используется подстроечный резистор, подключённый к свободному аналоговому входу [Arduino](#), показание АЦП которой как раз лежат в диапазоне от 0 до 1023. Например, `calibrationDC(analogRead(A5));` // Показания возвращаемые функцией `readDC()` корректируются потенциометром подключённым к выводу A5.

Напряжение на выходе модуля «S» может измениться при подключении модуля к цепи измеряемого тока, даже если цепь обесточена. На это могут повлиять такие факторы как: токи утечки, электромагнитные и магнитные поля создаваемые другими цепями, или устройствами, намагниченные предметы вблизи модуля и т.д. Значит, перед началом работы с модулем, нужно указать напряжение на выходе «S» которое соответствует отсутствию тока в подключённой цепи. Это напряжение указывается через функцию `setZeroVDC()` или `setZeroVAC()`, например, `setZeroVDC(2.5);` // напряжение на выходе модуля при отсутствии измеряемого постоянного тока соответствует значению 2,5 В. Так же эта функция может быть использована если Вам требуется измерить не реальное значение тока, а отклонение показаний от требуемой силы тока.

Получить среднее значение напряжения на выходе «S» модуля можно вызвав функцию `getZeroVDC()` или `getZeroVAC()`, например, `float i = getZeroVDC();` // сохранить среднее значение напряжения в переменную `i`. Функция вернёт усреднённое напряжение в вольтах на выходе модуля, которое соответствует текущему току в измеряемой цепи. Далее это значение можно использовать в качестве параметра для функции `setZeroVDC()` или `setZeroVAC()`.

Если Вы желаете использовать модуль для измерения тока в разных цепях, то в коде `setup` можно указать: `float i = getZeroVDC();` `setZeroVDC(i);` // получить и установить среднее значение напряжения на выходе модуля, например, соответствующее отсутствию постоянного тока в измеряемой цепи. Но тогда при каждом старте скетча, ток в цепи действительно должен отсутствовать.

Если Вы желаете использовать модуль для измерения тока в одной и той же цепи, то можно считать напряжение на выходе модуля (при отсутствии измеряемого тока) используя функцию `getZeroVDC()` или `getZeroVAC()`, только один раз. Запомнить полученное значение. И переписать скетч, где в коде `setup` вызывать только функцию `setZeroVDC()` или `setZeroVAC()` с указанием этого значения, не вызывая функцию `getZeroVDC()` или `getZeroVAC()`. Например, `setZeroVDC(2.5)`; Тогда обесточивать цепь при старте скетча не потребуется.

Подробнее про установку библиотеки читайте в нашей [инструкции](#).

Примеры:

Измерение постоянного тока:

```
// // Измерение постоянного тока:
#include <iarduino_ACS712.h> // Подключаем библиотеку для работы с Трема датчиками тока.
iarduino_ACS712 sensor(A0); // Объявляем объект sensor для работы с Трема датчиком тока,
// указывая что он подключён к выводу A0.

void setup(){ //
  Serial.begin(9600); // Иницируем передачу данных в монитор на скорости 9600 бод.
  float v=sensor.getZeroVDC(); // Считываем усреднённое напряжение с выхода модуля при отсутствии измеряемого тока (I = 0 A)
  sensor.setZeroVDC(v); // Указываем чему равно напряжение на выходе модуля при отсутствии измеряемого тока (I = 0 A)
// В данном случае, при старте скетча, измеряемая цепь должна быть обесточена!
// Но можно сделать по другому:
// Однократно считать напряжение функцией getZeroVDC(), при отсутствии измеряемого тока и зап
// После чего исключить функцию getZeroVDC() из скетча.
// sensor.setZeroVDC(2.49); // А в качестве аргумента функции setZeroVDC() указать запомненное ранее значение, например,
// Тогда, при старте, обесточивать измеряемую цепь не требуется.
// sensor.calibrationDC(511); // Если показания тока не соответствуют действительности, то ...
// их можно линейно уменьшить или увеличить, вызвав функцию calibrationDC() с параметром.
} // Параметр функции calibrationDC() - целое число от 0 до 1023 (по умолчанию 511).
// Чем выше значение параметра, тем выше показания тока и наоборот.

void loop(){ //
  float i = sensor.readDC(); // Считываем силу постоянного тока в переменную i. Можно считывать силу тока от -5 А до +5 А.
// Функция readDC() автоматически сглаживает показания силы тока (чем чаще она вызывается, те
// Но данную функцию можно вызвать с параметром в виде целого числа от 1 до 100, указав требу
```

```

// Чем выше значение параметра, тем сильнее сглаживание показаний (тем плавнее будут изменять
Serial.println(i); // Выводим считанную силу тока в монитор последовательного порта.
Serial.println(" A."); // Добавляем единицу измерений после выведенного значения.
delay(500); // Устанавливаем задержку в 500 мс (не обязательно).
} //
// В этом скетче описаны все функции для работы с постоянным током: getZeroVDC(), setZeroVDC()
// Для работы с переменным током используются похожие функции: getZeroVAC(), setZeroVAC()

```

В данном скетче, [Трема датчик тока 5A](#) используется для вывода измеренного постоянного тока в монитор последовательного порта. Следующий скетч предназначен для измерения переменного тока и отличается от данного скетча только тем, что в нём изменены окончания функций библиотеки [iarduino_ACS712](#) (вместо DC указано AC).

Измерение переменного тока:

```

// // Измерение переменного тока:
#include <iarduino_ACS712.h> // Подключаем библиотеку для работы с Трема датчиками тока.
iarduino_ACS712 sensor(A0); // Объявляем объект sensor для работы с Трема датчиком тока,
// указывая что он подключён к выводу A0.
void setup(){ //
  Serial.begin(9600); // Иницируем передачу данных в монитор на скорости 9600 бод.
  float v=sensor.getZeroVAC(); // Считываем усреднённое напряжение с выхода модуля при отсутствии измеряемого тока (I = 0 A)
  sensor.setZeroVAC(v); // Указываем чему равно напряжение на выходе модуля при отсутствии измеряемого тока (I = 0 A)
// В данном случае, при старте скетча, измеряемая цепь должна быть обесточена!
// Но можно сделать по другому:
// Однократно считать напряжение функцией getZeroVAC(), при отсутствии измеряемого тока и записать его в переменную v
// После чего исключить функцию getZeroVAC() из скетча.
// sensor.setZeroVAC(0.25); // А в качестве аргумента функции setZeroVAC() указать запомненное ранее значение, например,
// Тогда, при старте, обесточивать измеряемую цепь не требуется.
// sensor.calibrationAC(511); // Если показания тока не соответствуют действительности, то ...
// их можно линейно уменьшить или увеличить, вызвав функцию calibrationAC() с параметром.
} // Параметр функции calibrationAC() - целое число от 0 до 1023 (по умолчанию 511).

```



```

// Чем выше значение параметра, тем выше показания тока и наоборот.
void loop(){
  float i = sensor.readAC(); // Считываем силу переменного тока в переменную i. Можно считывать силу тока от 0 до +5 А.
  // Функция readAC() автоматически сглаживает показания силы тока (чем чаще она вызывается, тем
  // Но данную функцию можно вызвать с параметром в виде целого числа от 1 до 100, указав требу
  // Чем выше значение параметра, тем сильнее сглаживание показаний (тем плавнее будут изменять
  Serial.print(i); // Выводим считанную силу тока в монитор последовательного порта.
  Serial.println(" A."); // Добавляем единицу измерений после выведенного значения.
  delay(500); // Устанавливаем задержку в 500 мс (не обязательно).
}
//
// В этом скетче описаны все функции для работы с переменным током: getZeroVAC(), setZeroVAC(
// Для работы с постоянным током используются похожие функции: getZeroVDC(), setZeroVDC(

```

В данном скетче, [Trema датчик тока 5A](#) используется для вывода измеренного переменного тока в монитор последовательного порта. Данный скетч отличается от предыдущего только тем, что в нём изменены окончания функций библиотеки [iarduino_ACS712](#) (вместо DC указано AC).

Измерение переменного тока с регулировкой показаний:

Как видно из предыдущих скетчей, функция `calibrationDC()` или `calibrationAC()`, позволяет откорректировать выводимые показания, но её удобно использовать с подстроечным резистором подключённым к свободному аналоговому входу [Arduino](#), что и сделано в следующем скетче:

```

// // Измерение переменного тока с регулировкой показаний:
#include <iarduino_ACS712.h> // Подключаем библиотеку для работы с Trema датчиками тока.
iarduino_ACS712 sensor(A0); // Объявляем объект sensor для работы с Trema датчиком тока, указывая что он подключён
const uint8_t pinRes = A3; // Определяем вывод к которому подключён потенциометр для регулировки показаний.
//
void setup(){
  Serial.begin(9600); // Иницируем передачу данных в монитор на скорости 9600 бод.
  sensor.setZeroVAC(0.00332f); // Указываем чему равно напряжение на выходе модуля,

```

```

// при отсутствии измеряемого переменного тока (I = 0 A).
// Serial.print(sensor.getZeroVAC()); // Вместо значения 0.00332 В, Вам нужно указать то значение,
// Serial.println(" B."); // которое вернёт функция getZeroVAC() при отсутствии тока в измеряемой цепи.
} // В данном случае, при старте скетча, измеряемую цепь НЕ требуется обесточивать.
void loop(){ //
    int k=analogRead(pinRes); // Считываем показания с потенциометра в переменную k, её значением будет целое число от 0 до 1023.
    sensor.calibrationAC(k); // Выполняем калибровку показаний Трема датчика тока,
    // вызвав функцию calibrationAC() с параметром в виде целого числа от 0 до 1023.
    float i=sensor.readAC(); // Считываем силу переменного тока в переменную i. Можно считывать силу тока от 0 до +5 А.
    // Функция readAC() автоматически сглаживает показания силы тока.
    Serial.print(i); // Выводим считанную силу тока в монитор последовательного порта.
    Serial.println(" A."); // Добавляем единицу измерений после выведенного значения.
    delay(500); // Устанавливаем задержку в 500 мс (не обязательно).
} //

```

В данном скетче, показания силы тока, можно корректировать поворотом потенциометра. Центральное положение потенциометра будет означать отсутствие корректировки, а его поворот влево или вправо, будет линейно уменьшать или увеличивать показания. Аналогичным образом можно создать скетч для измерения постоянного тока с регулировкой показаний. Все представленные скетчи имеются в примерах библиотеки [iarduino_ACS712](#).

Описание основных функций библиотеки:

Библиотека [iarduino_ACS712](#) предназначена для работы только с [Трема датчиком тока 5А](#).

Подключение библиотеки:

```

#include <iarduino_ACS712.h> // Подключаем библиотеку для работы с Трема датчиками тока.
iarduino_ACS712 sensor(ВЫВОД); // Объявляем объект sensor указывая к какому аналоговому выводу подключён Трема датчик тока.

```

Функции readDC(); / readAC;

- Назначение: Чтение силы тока (DC - постоянного, AC - переменного).
- Синтаксис:
 - `readDC([КОЭФФИЦИЕНТ]);` // чтение постоянного тока.
 - `readAC([КОЭФФИЦИЕНТ]);` // чтение переменного тока.
- Параметры:
 - КОЭФФИЦИЕНТ - необязательное целое число от 1 до 100 определяющее степень сглаживания показаний.
- Возвращаемые значения:
 - `float` - сила тока в амперах.
- Примечание:
 - Если функция вызвана без параметра, то сглаживание показаний осуществляется автоматически. Чем чаще вызывается функция, тем сильнее сглаживание.
 - Если функция вызвана с параметром, то чем выше его значение, тем сильнее сглаживание показаний. Чем сильнее сглаживание показаний, тем плавнее они меняются.
 - Если функция вызвана с параметром 1, то показания выводятся без сглаживания.
- Пример:

```
float IDC = sensor.readDC(); // Читаем постоянный ток.  
float IAC = sensor.readAC(); // Читаем переменный ток.
```

Функции `calibrationDC()`; / `calibrationAC`;

- Назначение: Корректировка показаний модуля (DC - для постоянного тока, AC - для переменного тока).
- Синтаксис:
 - `calibrationDC(КОЭФФИЦИЕНТ);` // корректировка показаний постоянного тока.
 - `calibrationAC(КОЭФФИЦИЕНТ);` // корректировка показаний переменного тока.
- Параметры:
 - КОЭФФИЦИЕНТ - целое число от 0 до 1023 (по умолчанию 511), определяющее отклонение показаний.
- Возвращаемые значения: нет
- Примечание:

- Если функция вызвана с параметром 511, то чтение тока будет осуществляться без корректировок.
- Если функция вызвана с параметром меньше 511, то значение выводимого тока будет пропорционально меньше.
- Если функция вызвана с параметром выше 511, то значение выводимого тока будет пропорционально больше.
- Диапазон коэффициента выбран в пределах от 0 до 1023 не случайно. Числа в том же диапазоне возвращает функция чтения данных с аналоговых входов `analogRead(вывод)`, что позволяет корректировать показания датчика тока с помощью внешних подстроечных или переменных резисторов.
- Пример:

```
float IDC = sensor.calibrationDC(200); // Указываем что функция readDC() должна возвращать значение постоянного тока меньше чем 200 мА
float IAC = sensor.calibrationAC(700); // Указываем что функция readAC() должна возвращать значение переменного тока больше чем 700 мА
```

Функции `setZeroVDC()`; / `setZeroVAC()`;

- Назначение: Установка коррекции смещения нуля для (DC - для постоянного тока, AC - для переменного тока).
- Синтаксис:
 - `setZeroVDC(НАПРЯЖЕНИЕ);` // указываем напряжение на выходе модуля, соответствующее отсутствию постоянного тока.
 - `setZeroVAC(НАПРЯЖЕНИЕ);` // указываем напряжение на выходе модуля, соответствующее отсутствию переменного тока.
- Параметры:
 - НАПРЯЖЕНИЕ - число типа `float` определяющее напряжение на выходе модуля, которое соответствует току = 0 А.
- Возвращаемые значения: нет
- Примечание:
 - Данную функцию достаточно вызвать один раз в коде `setup`.
 - Параметр функции можно указать числом, а можно получить вызвав функцию `getZeroVDC()` или `getZeroVAC()` при отсутствии тока.
 - Данная функция является не обязательной, но она позволяет точно указать какому напряжению на выходе модуля, соответствует ток = 0 А, что увеличит точность показаний.
 - Если указать напряжение на выходе модуля, которое соответствует заданному току, а не отсутствию тока, то дальнейшие показания будут не реальными, а отклонениями от заданного тока.
- Пример:

```
sensor.setZeroVDC(2.49353f); // Указываем что отсутствию постоянного тока соответствует постоянное напряжение 2.49353 В на вых
sensor.setZeroVAC(0.00332f); // Указываем что отсутствию переменного тока соответствует переменное напряжение 0.00332 В на вых
```

Функции `getZeroVDC()`; / `getZeroVAC()`;

- Назначение: Чтение усреднённого напряжения на выходе модуля (DC - постоянного, AC - переменного).
- Синтаксис:
 - `getZeroVDC()`; // чтение усреднённого постоянного напряжения на выходе модуля.
 - `getZeroVAC()`; // чтение усреднённого переменного напряжения на выходе модуля.
- Параметры: нет.
- Возвращаемые значения:
 - `float` - усреднённое напряжение на выходе модуля в вольтах.
- Примечание:
 - Данная функция является не обязательной и требуется только для получения параметра к функции `setZeroVDC()` или `setZeroVAC()`.
 - Данная функция не измеряет ток или напряжение исследуемой цепи.
 - Если обесточить исследуемую цепь и вызвать функцию `setZeroVDC()` или `setZeroVAC()` с указанием значения возвращённого данной функцией, то дальнейшие показания тока будут более точными.
 - Если в исследуемой цепи установить требуемый ток и вызвать функцию `setZeroVDC()` или `setZeroVAC()` с указанием значения возвращённого данной функцией, то дальнейшие показания тока будут отклонением от требуемого.
- Пример:

```
sensor.setZeroVDC(sensor.getZeroVDC()); // Устанавливаем текущее напряжение на выходе модуля при котором текущий постоянный то
sensor.setZeroVAC(sensor.getZeroVAC()); // Устанавливаем текущее напряжение на выходе модуля при котором текущий переменный то
```

Применение:

- Амперметры, системы учёта электричества;

Ссылки:

