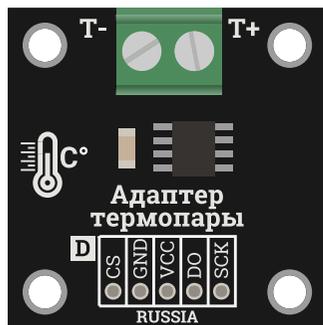


# Нормализатор сигнала термопары К-типа, max6675 (Трема-модуль)



## Общие сведения:

[Трема-модуль Адаптер термопары с термопарой К-типа](#) — позволяет измерять температуру объектов и окружающей среды. Адаптер позволяет измерять температуру в диапазоне от 0 до 800 °С (при температурах от 800 до 1023 °С точность показаний резко снижается), а корпус термопары рассчитан на температуру до 600 °С, значит и измерения необходимо производить в диапазоне от 0 до 600 °С. Адаптер можно использовать для измерения температуры различных нагревательных элементов: паяльники, печи, камины, бойлеры и т.д. Можно использовать для реализации газ-контроля в газовых плитах, горелках, бойлерах и т.д.

## Характеристики:

- Напряжение питания модуля (Vcc): 3,0 ... 5,5 В постоянного тока.
- Ток потребляемый модулем: до 1,5 мА.
- Уровень логической «1» на шине модуля: > 0,7 Vcc.
- Уровень логического «0» на шине модуля: < 0,3 Vcc.
- Диапазон измеряемых температур чипом: 0 ... 800 °С.
- Диапазон рабочих температур термопары: 0 ... 600 °С.
- Резьба термопары 6М x 1.0 мм (для крепления нужен ключ на 10).
- Длина кабеля термопары 0,5 м.

## Подключение:

Выводы термопары подключаются к клеммнику на модуле:

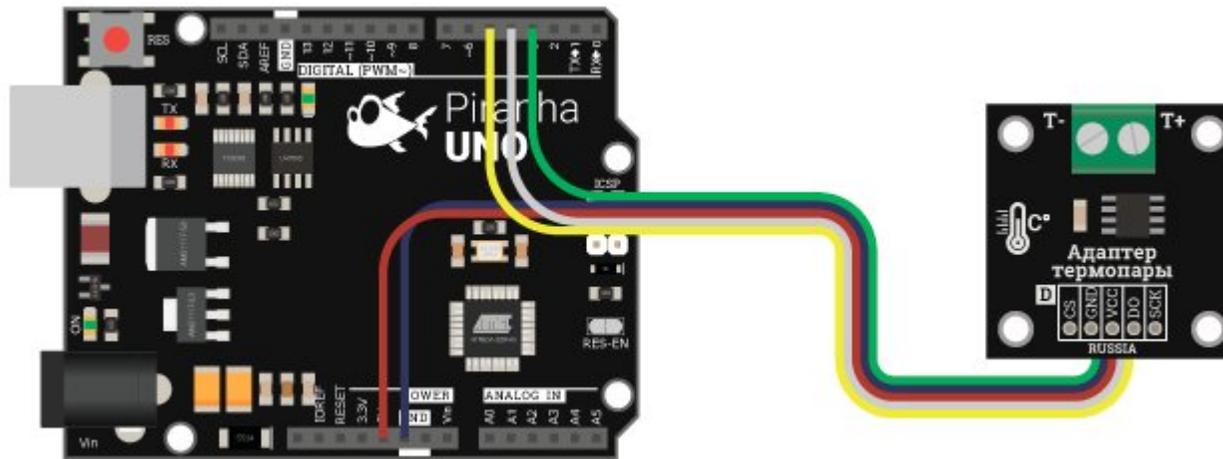
- **синий контакт** – термопары к выводу «Т-»;
- **красный контакт** – термопары к выводу «Т+»;

Выводы модуля DO, CS и CSK подключаются к любым выводам [Arduino](#), номера которых указываются в скетче при объявлении объекта [библиотеки max6675](#).

Модуль удобно подключать 3 способами, в зависимости от ситуации:

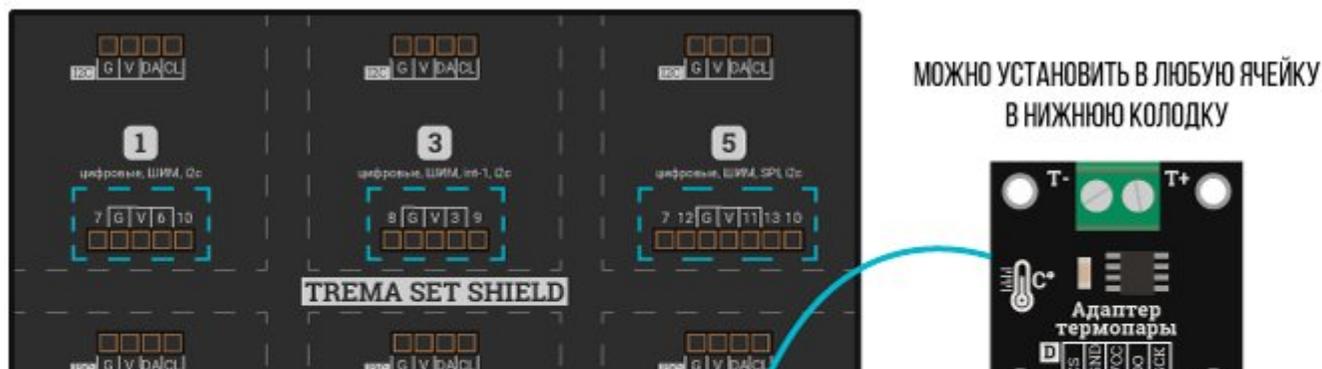
### Способ - 1 : Используя проводной шлейф и Piranha UNO

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



### Способ - 2 : Используя Trema Set Shield

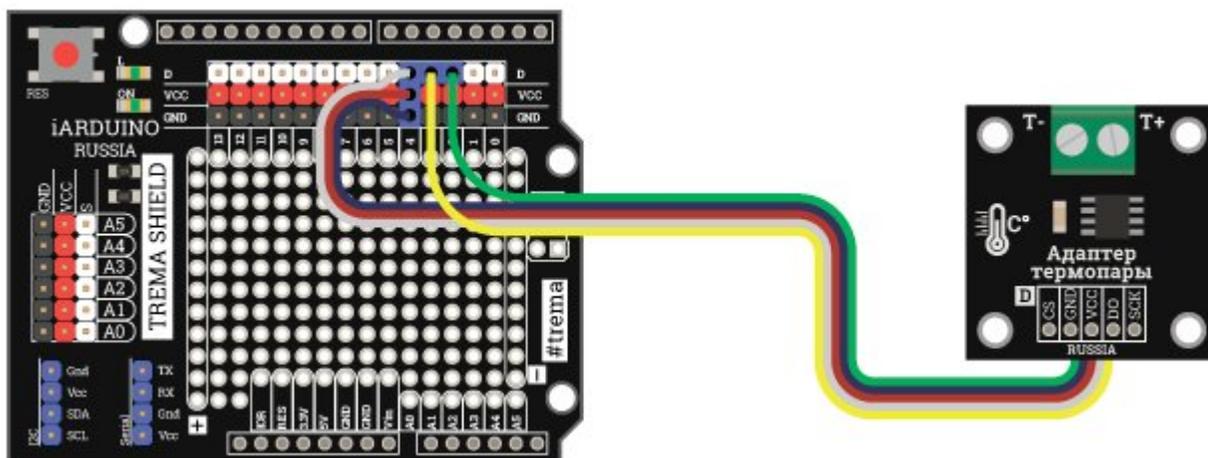
Модуль можно подключить к любому из цифровых или аналоговых входов Trema Set Shield.





### Способ - 3 : Используя проводной шлейф и Shield

Используя 3-х и 2-х проводной шлейфы, к Trema Shield, Trema-Power Shield, Motor Shield, Trema Shield NANO и тд.



### Питание:

Входное напряжение питания от 3,3 или 5 В подаётся на выводы Vcc и GND.

### Подробнее о датчике температуры:

В основу работы термопары заложен термоэлектрический эффект, это процесс возникновения ЭДС (электродвижущей силы) на спайке (соединении) разнородных металлов, интенсивность которой прямо пропорциональна температуре в месте этого соединения. Возникающая в термопаре ЭДС очень мала и её нельзя измерить используя аналоговые входы [Arduino](#), но с этой задачей справляется чип max6675. Сигнал с термопары сначала поступает на операционные усилители чипа, а уже потом на 12 битный АЦП, далее данные выводятся по шине SPI в виде двух байт, старшие 12 бит которых являются числом температуры. Для преобразования 12 битного числа в °C его нужно разделить на 4,

получается что температура выводится с разрешением 0,25 °C. Так как данные из чипа только читаются, без отправки данных чипу, то на его шине SPI отсутствует вывод MOSI. Немаловажным фактом является и то, что в чипе max6675 реализована функция компенсации холодного спая. Дело в том, что ЭДС возникает не только на спае металлов термопары, но и на соединении этих металлов с проводом, что без функции компенсации холодного спая влияло-бы на показания температуры.

Для работы с чипом max6675 рекомендуем воспользоваться одноимённой [библиотекой max6675](#), которая реализует получение данных чипа по программной шине SPI, значит, для подключения можно использовать любые выводы [Arduino](#).

## Пример:

### Вывод температуры в монитор последовательного порта в градусах Цельсия и Фаренгейта.

```
#include <max6675.h> // Подключаем библиотеку max6675 для работы с датчиком температуры
const uint8_t thermoDO = 4; // Определяем константу с указанием № вывода Arduino к которому подключён
const uint8_t thermoCS = 5; // Определяем константу с указанием № вывода Arduino к которому подключён
const uint8_t thermoCLK = 6; // Определяем константу с указанием № вывода Arduino к которому подключён
MAX6675 thermo(thermoCLK, thermoCS, thermoDO); // Объявляем объект thermo для работы с функциями и методами библиотеки m
//
void setup(){ //
  Serial.begin(9600); // Инициуем передачу данных в монитор последовательного порта на скорос
  Serial.println("Тест MAX6675"); //
  delay(500); // Ждём завершения переходных процессов датчика температуры после подачи
} //
//
void loop(){ //
  Serial.print ( "C = " ); //
  Serial.print ( thermo.readCelsius() ); // Выводим температуру в °C
  Serial.print ( ", F = " ); //
  Serial.print ( thermo.readFahrenheit() ); // Выводим температуру в °F
  Serial.println( "." ); //
  delay(1000); // Ждём 1 секунду
} //
```

В [библиотеке max6675](#) реализовано всего 2 функции: `readCelsius()` и `readFahrenheit()`. Они не принимают никаких параметров, а только возвращают температуру в виде числа типа `double`. Обе функции приведены в строках скетча 15 и 17.

При создании объекта нужно указать номера выводов [Arduino](#) к которым подключён модуль (вывод `SCK` , вывод `CS` , вывод `DO`), как это сделано в 5 строке скетча.

## Ссылки:

- [DataSheet](#);
- [Библиотека max6675](#) для работы с одноимённым чипом на модуле датчика температуры;