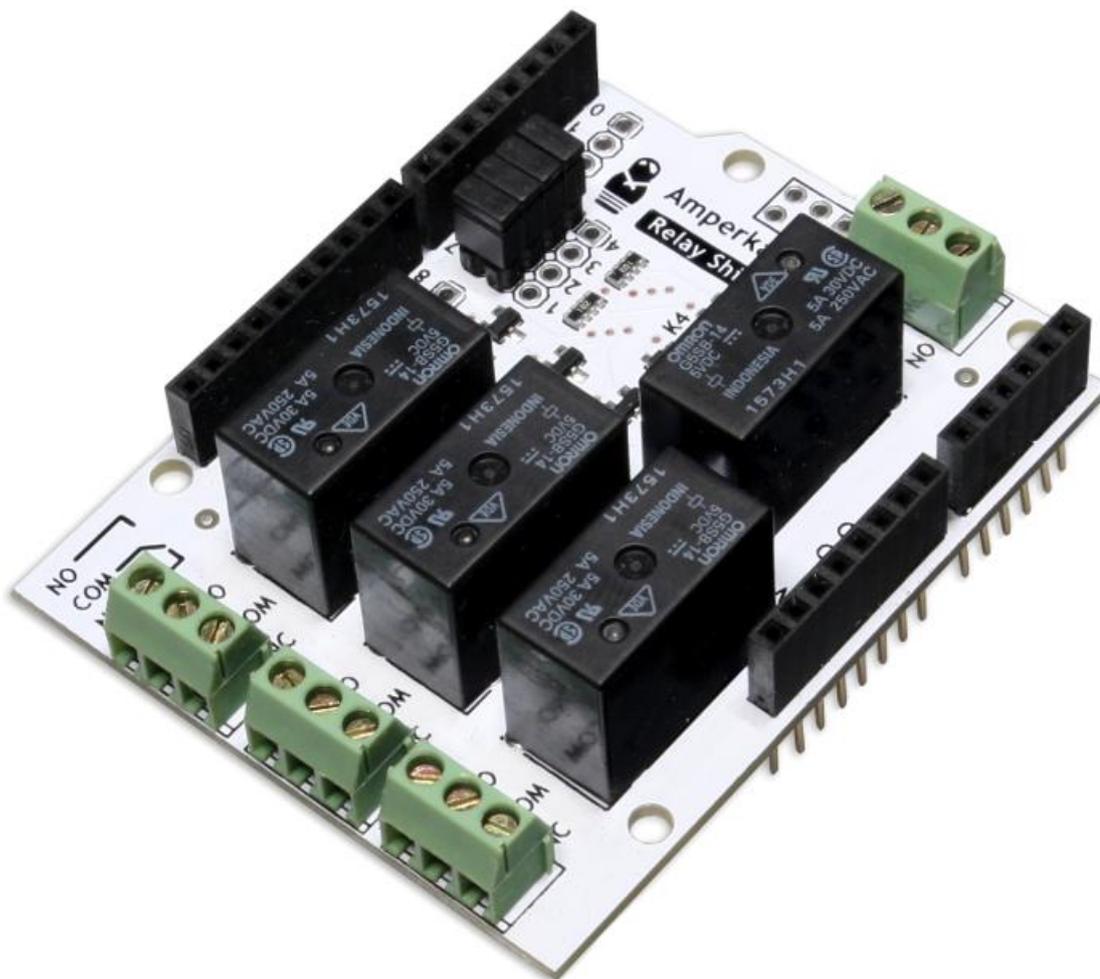


Relay Shield

Для управления электроприборами, которые питаются от бытовой электросети, люди пользуются различными клавишными выключателями и тумблерами. Чтобы управлять такими электроприборами с помощью микроконтроллера существует специальный тип выключателей - электромеханические реле. Relay Shield содержит четыре таких реле и позволяет Arduino управлять четырьмя электроприборами.



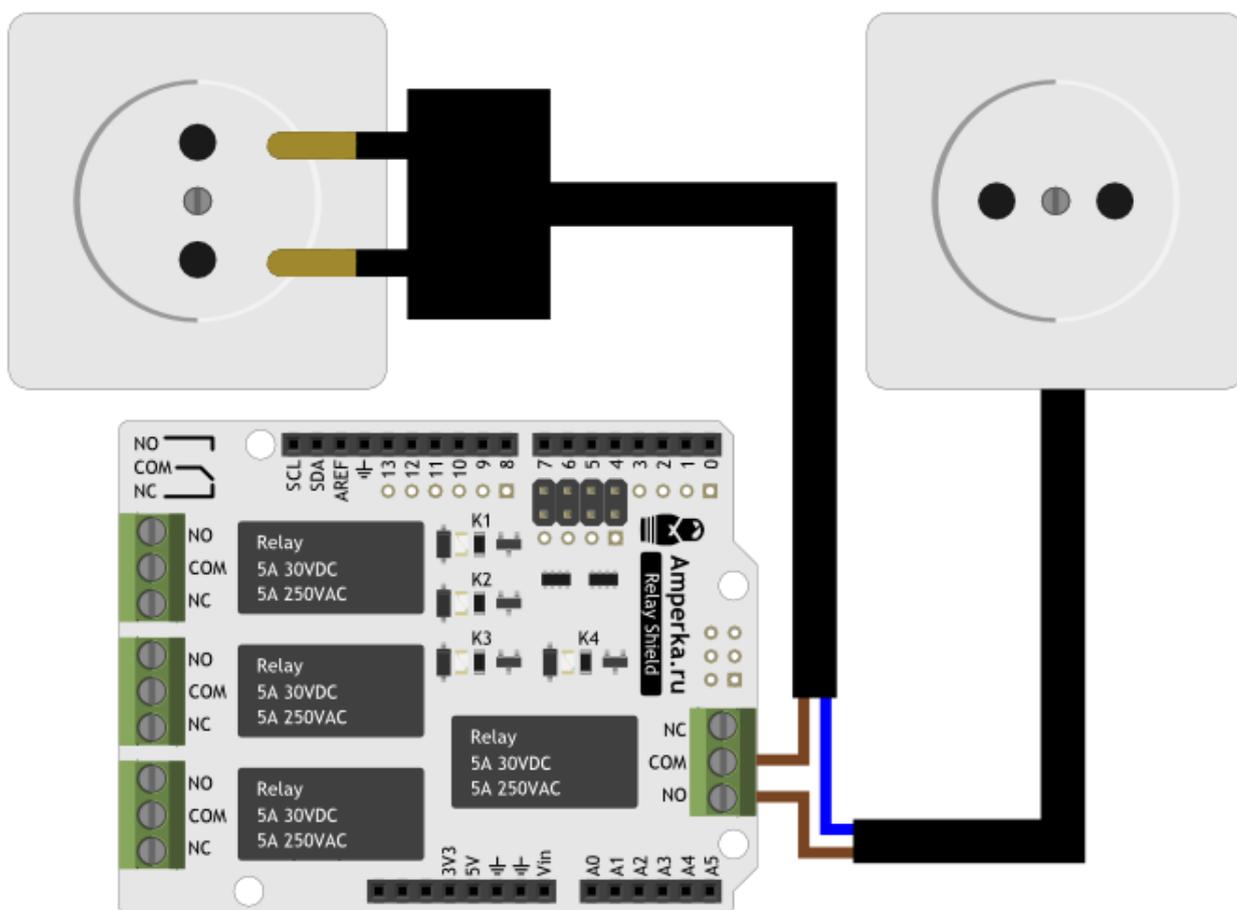
Внимание!

Работа с высоким напряжением опасна для вашего здоровья и жизни. На плате существуют области, прикосновение к которым приведёт к поражению электрическим током. Это винты контактных колодок и места пайки выводов контактных колодок и реле. Не работайте с платой, если она подключена к бытовой сети. Для готового устройства используйте изолированный корпус.

Если вы сомневаетесь как подключить к реле электроприбор, работающий от общей сети 220 В и у вас есть сомнения, вопросы на тему того как это делается, остановитесь: вы можете устроить пожар или убить себя. Убедитесь, что у вас в голове - кристальное понимание принципа работы реле и опасностей, которые связаны с высоким напряжением.

Источник напряжения

Управляемая нагрузка



Используемые пины

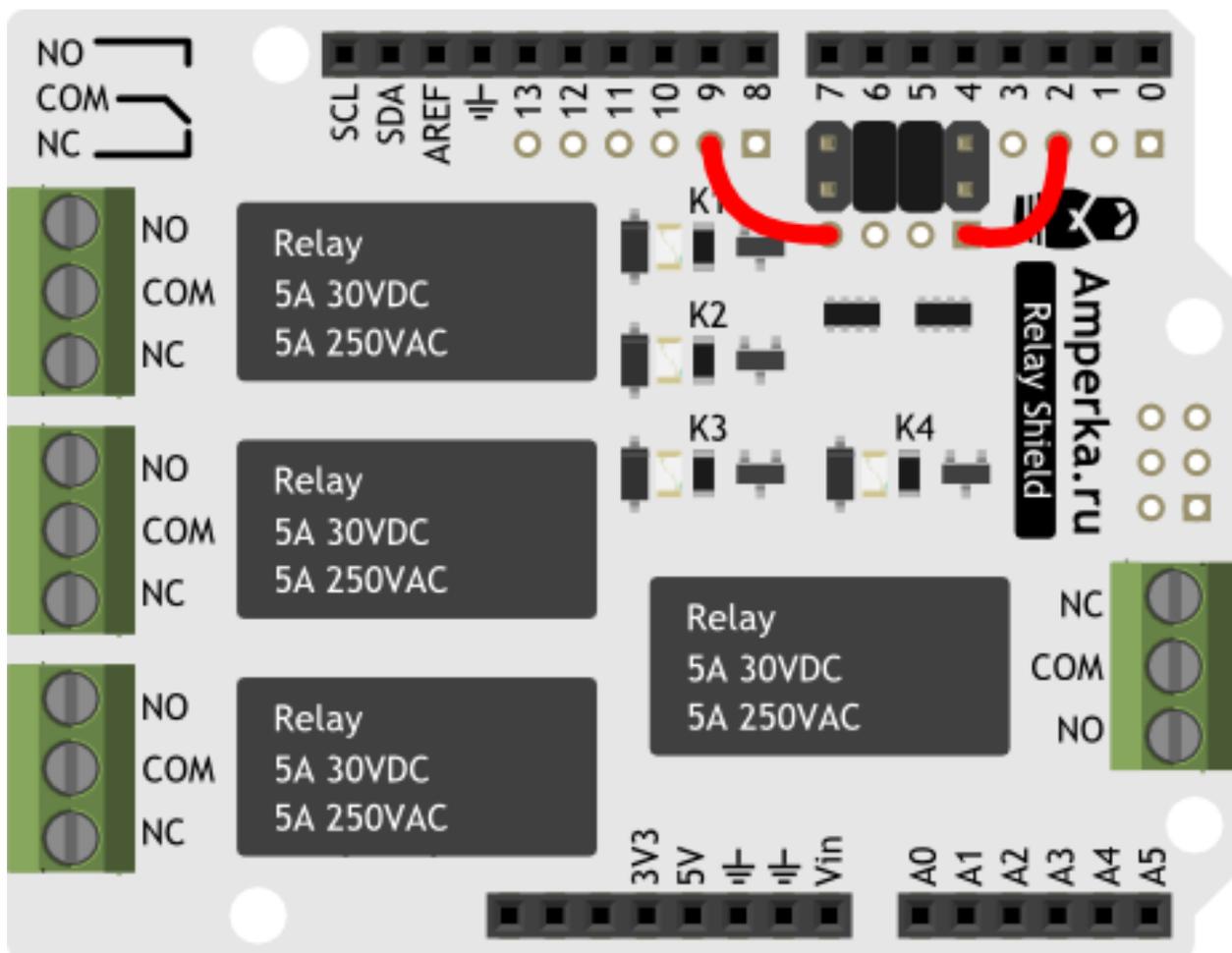
Для управления реле используются контакты 4, 5, 6 и 7 Arduino.

Реле	Контакт Arduino
Реле 1	7
Реле 2	6
Реле 3	5
Реле 4	4

При установке логической единицы на контакте Arduino срабатывает соответствующее реле. При этом напряжение логической единицы может быть как 5В, так и 3.3В. При подаче на контакт Arduino логического нуля или при исчезновении напряжения на Arduino, реле возвращается в нормальное положение.

Контакты выбора управляющих пинов

Контакты 4, 5, 6 и 7, используемые для управления реле, подключены к вспомогательной логике управления реле через джамперы. Если в вашем устройстве какие-то из этих пинов уже заняты (например, эти же пины используются для управления Motor Shield), вы можете использовать любой свободный цифровой пин для управления реле. Для этого необходимо снять джампер напротив занятого пина, и припаять проводок между луженым отверстием рядом со снятым джампером и луженым отверстием рядом с нужным пином.



На этой картинке мы перекинули управление первым реле с контакта 7 на контакт 9, а управление четвёртым реле с контакта 4 на контакт 2.

Индикатор состояния и обвязка реле

Микроконтроллер не может напрямую управлять реле: оно потребляет слишком большой ток и порождает выбросы обратного напряжения при отключении. Поэтому каждое реле подключено к управляющим контактам через транзистор, а обратный диод защищает остальную схему от выбросов напряжения. Между транзистором и обратным диодом находится индикатор состояния реле - светодиод. Светодиод горит если на реле подано напряжение. Если реле находится в нормальном состоянии - светодиод не горит.

Ограничения питания логической части

Отдельное реле в замкнутом состоянии потребляет 80мА из логической цепи в 5 вольт. Все 4 реле при одновременном включении потребляют 320мА. Если этот сценарий возможен в вашем проекте, необходимо удостовериться, что необходимый ток доступен.

Иными словами, все реле одновременно могут не работать в одном из следующих случаев.

- Вы питаете Arduino от USB-порта с пределом по току в 200мА, например, от разветвителя в клавиатуре. Используйте полноценный USB 2.0 или USB 3.0, чтобы обеспечить стабильное питание от USB

- Вы питаете Arduino внешним источником питания с высоким входным напряжением. Несмотря на то, что линейный регулятор напряжения на плате Arduino выдаёт до 800мА, их можно получить только, если обеспечена температура регулятора в 25 °С. Излишек напряжения линейный регулятор превращает в рассеиваемое тепло, компонент нагревается, предельный ток снижается. В этом случае используйте либо источник питания на 7-8 вольт вместо 8+, либо установите радиатор на регулятор напряжения, либо подавайте ровные 5 вольт непосредственно на пины 5V и GND или в порт USB.

Пример использования

[relayClick.ino](#)

```
//Определяем на каких пинах находятся реле
#define RELAY_1 7
#define RELAY_2 6
#define RELAY_3 5
#define RELAY_4 4

void setup() {
  // Конфигурируем нужные пины на выход
  for (int i = 4; i <= 7; ++i)
  {
    pinMode(i, OUTPUT);
  }
}

void loop() {

  //Включаем реле 1 на 5 секунд
  digitalWrite(RELAY_1, HIGH);
  delay(5000);
  //Отключаем реле 1
  digitalWrite(RELAY_1, LOW);

  //через секунду включаем реле 2 на 5 секунд
  delay(1000);

  digitalWrite(RELAY_2, HIGH);
  delay(5000);
  digitalWrite(RELAY_2, LOW);

  //Повторим с оставшимися реле то же самое
  delay(1000);

  digitalWrite(RELAY_3, HIGH);
  delay(5000);
  digitalWrite(RELAY_3, LOW);

  delay(1000);

  digitalWrite(RELAY_4, HIGH);
  delay(5000);
  digitalWrite(RELAY_4, LOW);

  delay(1000);
}
```