

# Соленоиды TAU-05...

Используйте соленоиды TAU-0520T или TAU-0530T для изготовления электрозамков, музыкальных инструментов и даже пушки Гаусса



## Общие сведения

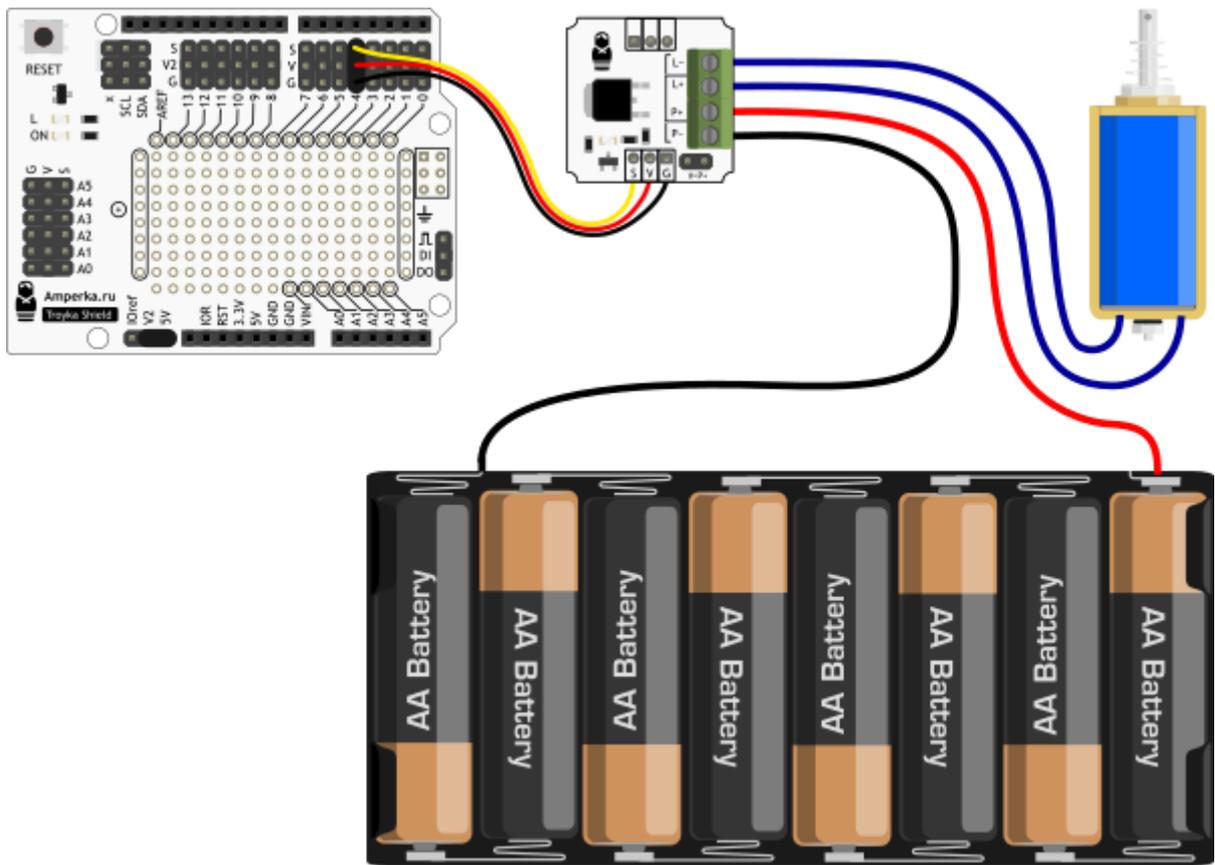
Соленоид — это цилиндрическая обмотка, длина которой значительно превышает ее диаметр. Слово соленоид образовано сочетанием двух слов — «solen» и «eidos», первое из которых переводится как труба, второе — подобный. То есть соленоид — это катушка, по форме напоминающая трубу.

Соленоиды — обычная катушка индуктивности, внутри которой при подаче напряжения возникает магнитное поле. Это поле втягивает в катушку магнитный сердечник, который и совершает механическую работу, например открывает замок или меняет положение клапана.

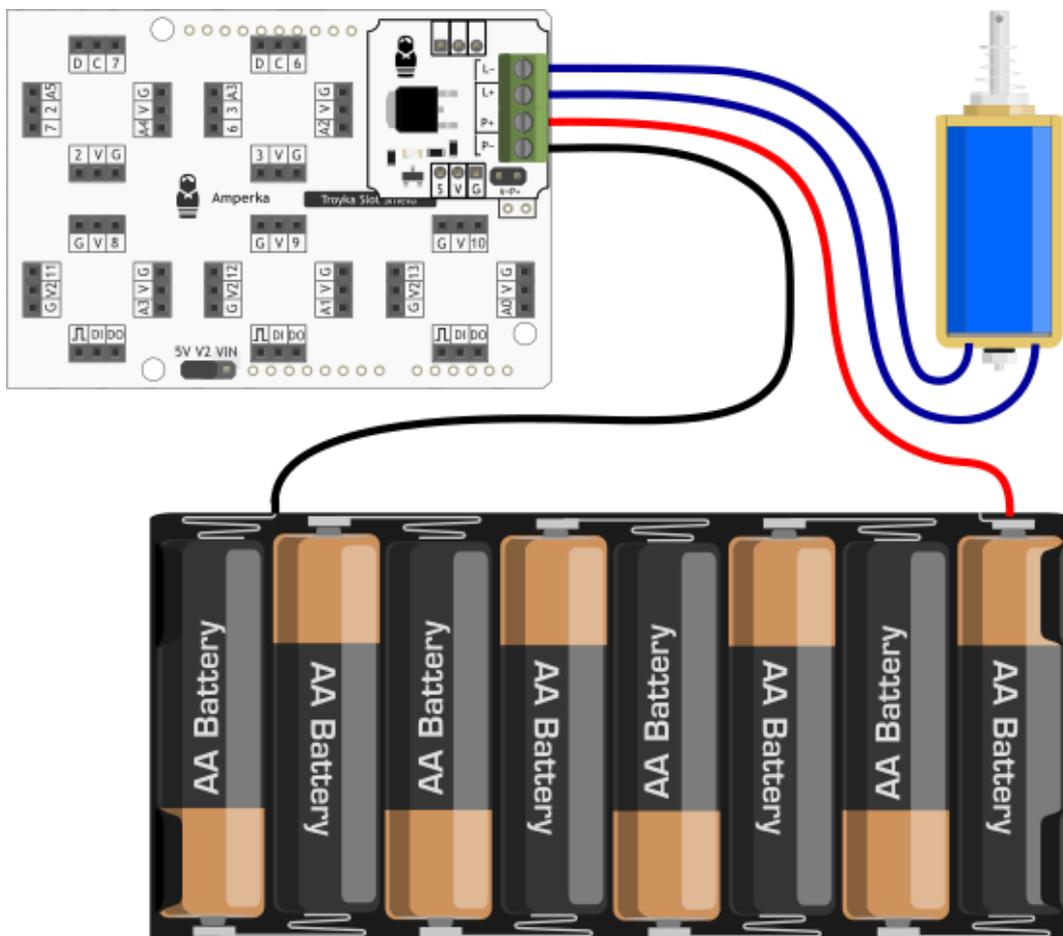
## Подключение и настройка

Соленоид срабатывает при появлении на его обмотки напряжения 12 вольт, поэтому подключайте его к управляющей плате через силовой ключ или реле.

При подключении к Arduino или Iskra JS удобно использовать Troyka Shield.



С Troyka Slot Shield можно обойтись без лишних проводов.



## Примеры использования

Рассмотрим несколько примеров работы соленоидов.

### Соленоидный Blink

В качестве примера будем включать и выключать соленоид раз в секунду, подключенный через силовой ключ к 4 пину.

#### [solenoidTest.ino](#)

```
void setup()
{
  // настраиваем пин 4 в режим выхода
  pinMode(4, OUTPUT);
}

void loop()
{
  // подаём на пин 4 «высокий сигнал»
  digitalWrite(4, HIGH);
  // ждём 1 секунду
  delay(1000);
  // подаём на пин 4 «низкий сигнал»
  digitalWrite(4, LOW);
  // ждём 1 секунду
  delay(1000);
}
```

### Электромеханический ксилофон

Автоматизируем детский музыкальный ксилофон.

#### *Как собрать*

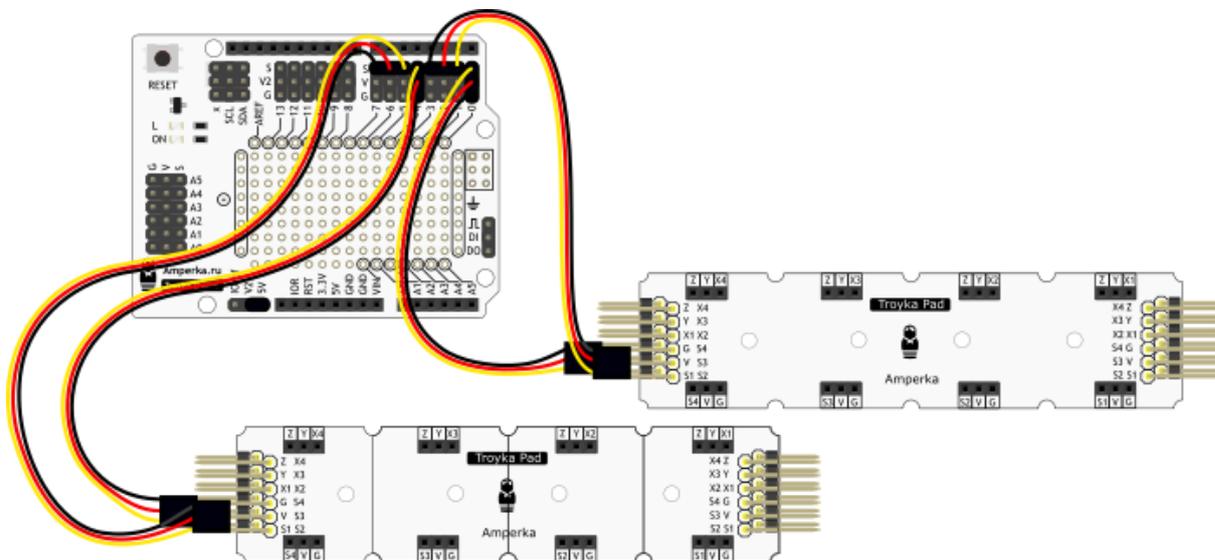
В качестве источника звука возьмём детский ксилофон на семь нот.



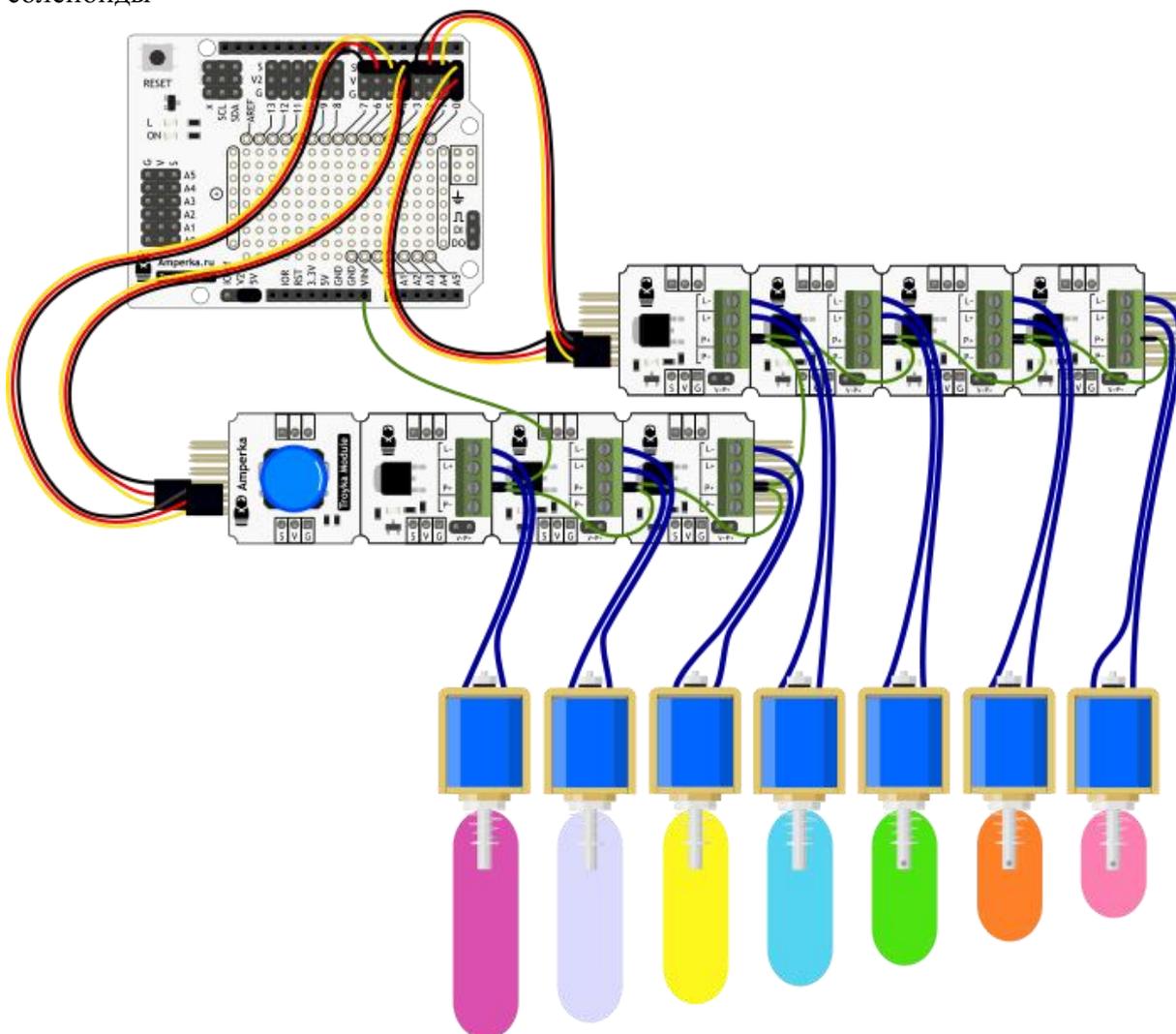
Соленоиды будут играть роль ударной палочки, следовательно понадобится семь

соленоидов и семь силовых ключей. Ключи подключим с помощью двух Troyka Pad 1×4 (Тройка-модуль) к управляющей плате Iskra Neo через Troyka Shiled.

Подключим Troyka Pad 1×4 к Troyka Shiled



После этого установим силовые ключи и кнопку на Troyka Pad 1×4 и подключим к ним соленоиды



Подключим 12 В через разъем внешнего питания к плате Iskra Neo.

### Принцип работы

Контроллер понимает мелодии в формате [рингтонов Nokia RTTTL](#) и конвертирует их в семиотную мелодию.

### Исходный код

#### [musicSolenoid.ino](#)

```
// пины соленоидов
#define SOLENOID_1_PIN    0
#define SOLENOID_2_PIN    1
#define SOLENOID_3_PIN    2
#define SOLENOID_4_PIN    3
#define SOLENOID_5_PIN    4
#define SOLENOID_6_PIN    5
#define SOLENOID_7_PIN    6

// пин кнопки
#define BUTTON_PIN        7

// массив пинов соленоидов
int solenoids[] = {SOLENOID_1_PIN, SOLENOID_2_PIN, SOLENOID_3_PIN,
                   SOLENOID_4_PIN, SOLENOID_5_PIN, SOLENOID_6_PIN,
                   SOLENOID_7_PIN};
// мелодия в формате RTTTL
unsigned char melodyRTTTL[] =
"Impossible:d=16,o=5,b=100:32d,32d#,32d,32d#,32d,32d#, "
"32d,32d#,32d,32d,32d#,32e,32f,32f#,32g,g,8p,g,8p,a#,p, "
"с6,p,g,8p,g,8p,f,p,f#,p,g,8p,g,8p,a#,p,с6,p,g,8p,g,8p, "
"f,p,f#,p,a#,g,2d,32p,a#,g,2с#,32p,a#,g,2с,p,a#4,с";

// вычисление размера массива при компиляции
const int N = sizeof(melodyRTTTL) / sizeof(char);

// имя мелодии
char name[12];
// длительность ноты по умолчанию
char duration[8];
// октава по умолчанию
char octave[8];
// темп / скорость в минуту
char bpm[8];

char melodyNotes[N];
int melodyNotesInt[N];

void setup()
{
  // открываем Serial-порт
  Serial.begin(9600);
  // назначаем пины соленоидов в режиме выхода
  for (int i = 0; i < 7; i++) {
    pinMode(solenoids[i], OUTPUT);
  }
}

void loop()
```

```

{
  // если нажата кнопка
  if (!digitalRead(BUTTON_PIN)) {
    // парсим мелодию RTTTL
    if (parsingMelody()) {
      // если парсинг мелодии формата RTTTL прошёл успешно
      // выводим имя мелодии
      Serial.println("Parsing is OK");
      Serial.println(name);
      // выводим длительность мелодии по умолчанию
      Serial.println(duration);
      // выводим октаву мелодии по умолчанию
      Serial.println(octave);
      // выводим темп мелодии
      Serial.println(bpm);
      // длина массива мелодии нот
      int len = strlen(melodyNotes);
    // Serial.println(melodyNotes);
      // преобразуем ноты в числовые значения
      parsingIntNotes();
      // играем музыку соленоидами
      playSolenoids(len);
    } else {
      Serial.println("Error");
    }
  }
}

// парсинг мелодии формата RTTTL
bool parsingMelody() {
  int i = 0;
  int j = 0;
  // поиск названия мелодии
  while (melodyRTTTL[i] != ':') {
    name[j++] = melodyRTTTL[i++];
    if(j > 12) {
      return false;
    }
  }

  // пропуск «:»
  i++;
  name[j] = '\0';

  // поиск длительности ноты
  j = 0;
  while (melodyRTTTL[i] != ',') {
    duration[j++] = melodyRTTTL[i++];
    if(j > 4) {
      return false;
    }
  }

  // пропуск «,»
  i++;
  duration[j] = '\0';

  // поиск октавы
  j = 0;
  while (melodyRTTTL[i] != ',') {
    octave[j++] = melodyRTTTL[i++];
    if(j > 4) {
      return false;
    }
  }
}

```

```

    }
    // пропуск «,»
    i++;
    octave[j] = '\0';

    // поиск темпа мелодии
    j = 0;
    while (melodyRTTTL[i] != ':') {
        bpm[j++] = melodyRTTTL[i++];
        if(j > 5) {
            return false;
        }
    }
}

// пропуск «:»
i++;
bpm[j] = '\0';

// запись мелодии нот
j = 0;
while (melodyRTTTL[i] != '\0') {
    if (isalpha(melodyRTTTL[i])) {
        melodyNotes[j++] = melodyRTTTL[i++];
    } else {
        i++;
    }
}
melodyNotes[j] = '\0';
return true;
}

// функция преобразование буквенных нот мелодии: d, e, f, g, a, b, p
// в числовые значения от 0 до 7
void parsingIntNotes() {
    int i = 0;
    int j = 0;
    while (melodyNotes[i] != '\0') {
        if (melodyNotes[i] == 'c') {
            melodyNotesInt[j++] = 0;
        } else if (melodyNotes[i] == 'd') {
            melodyNotesInt[j++] = 1;
        } else if (melodyNotes[i] == 'e') {
            melodyNotesInt[j++] = 2;
        } else if (melodyNotes[i] == 'f') {
            melodyNotesInt[j++] = 3;
        } else if (melodyNotes[i] == 'g') {
            melodyNotesInt[j++] = 4;
        } else if (melodyNotes[i] == 'a') {
            melodyNotesInt[j++] = 5;
        } else if (melodyNotes[i] == 'b') {
            melodyNotesInt[j++] = 6;
        } else if (melodyNotes[i] == 'p') {
            melodyNotesInt[j++] = 8;
        }
        i++;
    }
}

void playSolenoids(int len) {
    for(int i = 0; i < len; i++) {
        int speed = 100;
        Serial.println(melodyNotesInt[i]);
        digitalWrite(solenoids[melodyNotesInt[i]], HIGH);
    }
}

```

```
    delay(speed);  
    digitalWrite(solenoids[melodyNotesInt[i]], LOW);  
    delay(speed);  
  }  
}
```

## Характеристики

### Соленоид TAU-0520T

- Рабочее напряжение: 12 В
- Потребляемый ток: до 300 мА
- Сопротивление: 21 Ом
- Ход штока: 10 мм
- Диаметр штока: 6 мм
- Габариты: 20×16×13 мм
- Вес: 20 г

### Соленоид TAU-0530T

- Рабочее напряжение: 12 В
- Потребляемый ток: до 1 А
- Сопротивление: 8 Ом
- Ход штока: 10 мм
- Диаметр штока: 6 мм
- Габариты: 30×16×13 мм
- Вес: 32 г

## Ресурсы

- [Векторное изображение соленоида TAU-0520T](#)
- [Векторное изображение соленоида TAU-0530T](#)