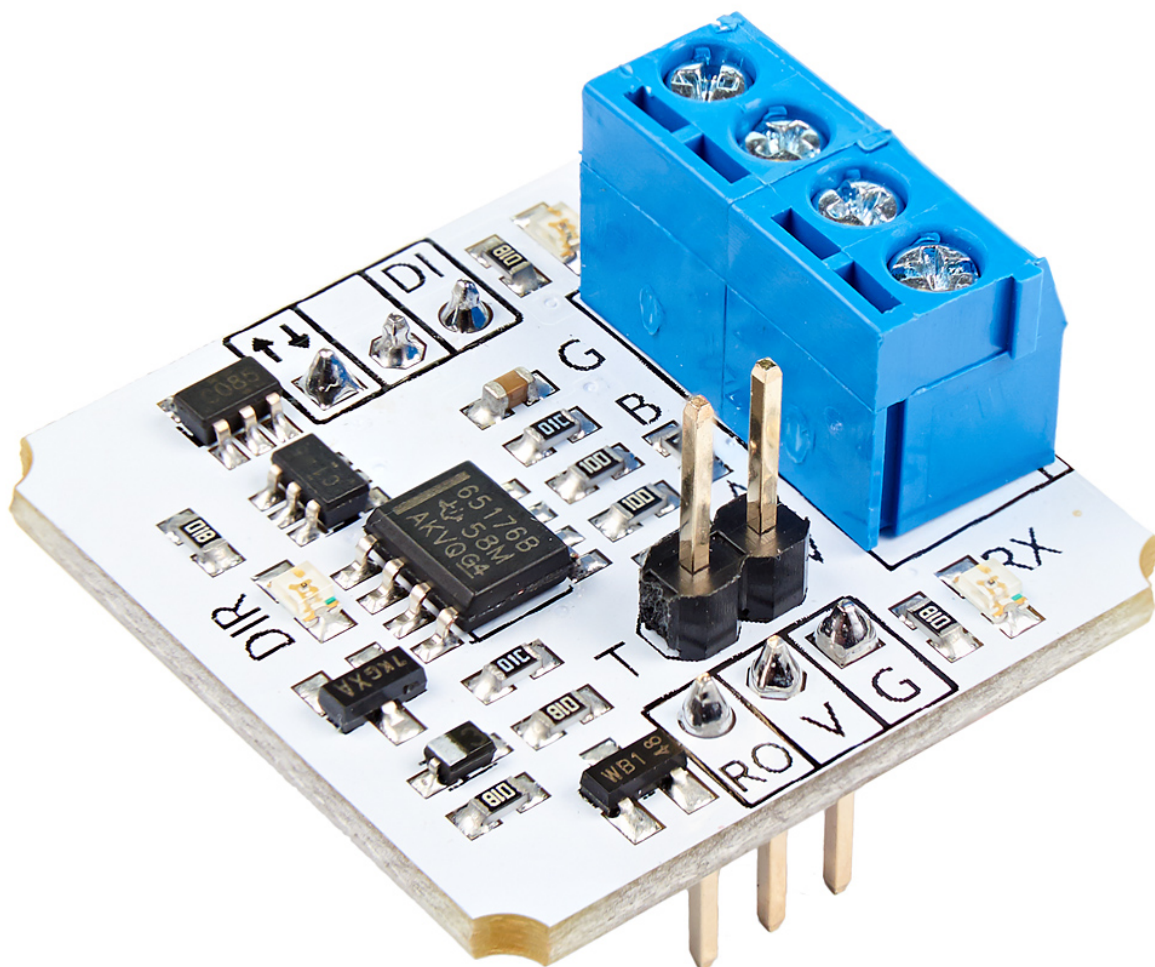


## RS-485 (Тройка-модуль)

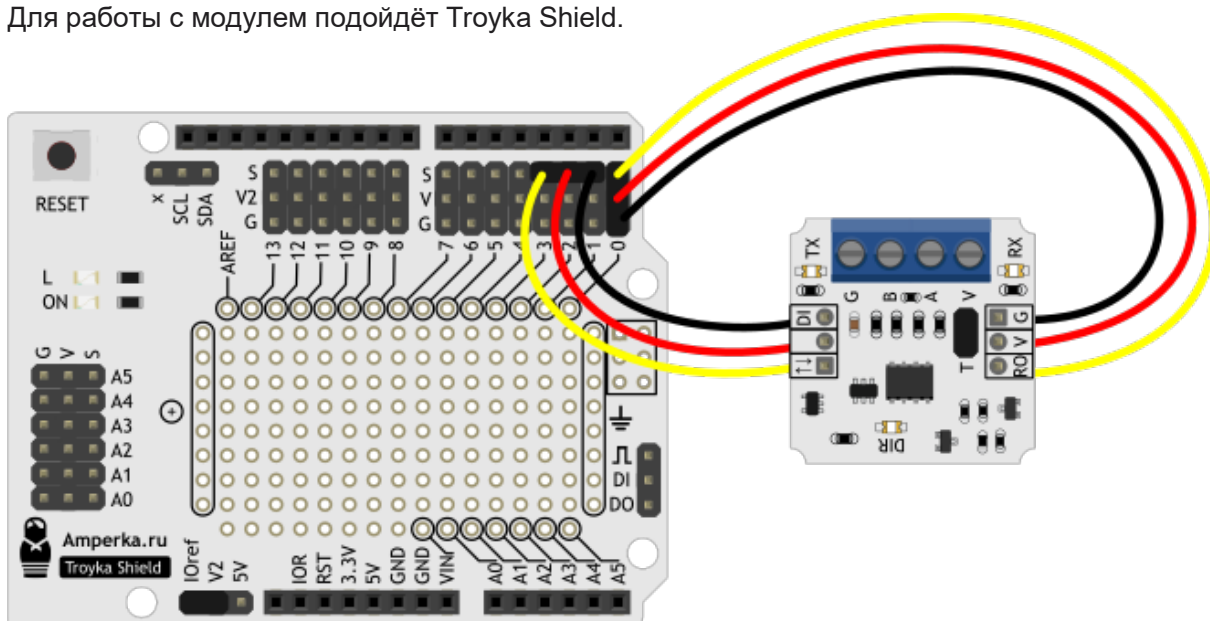
RS-485 (Тройка-модуль) поможет построить протяжённую проводную сеть из 32 устройств, используя всего два провода.



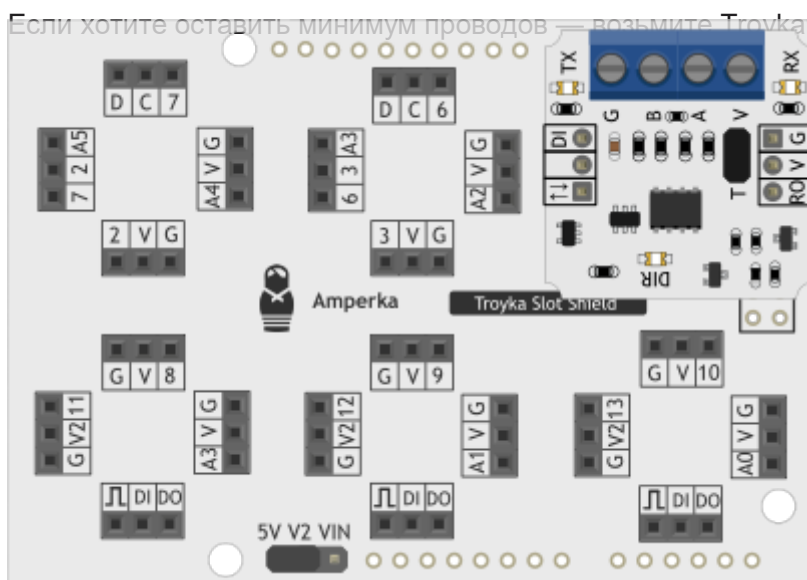
## Подключение и настройка

RS-485 (Тройка-модуль) общается с управляющей платой по протоколу UART через сигнальные пины **RO(TX)** и **DI(RX)**. Сигнальный пин **↑↓** служит для переключения модуля из режима приёмника в режим передатчика.

Для работы с модулем подойдёт Troyka Shield.



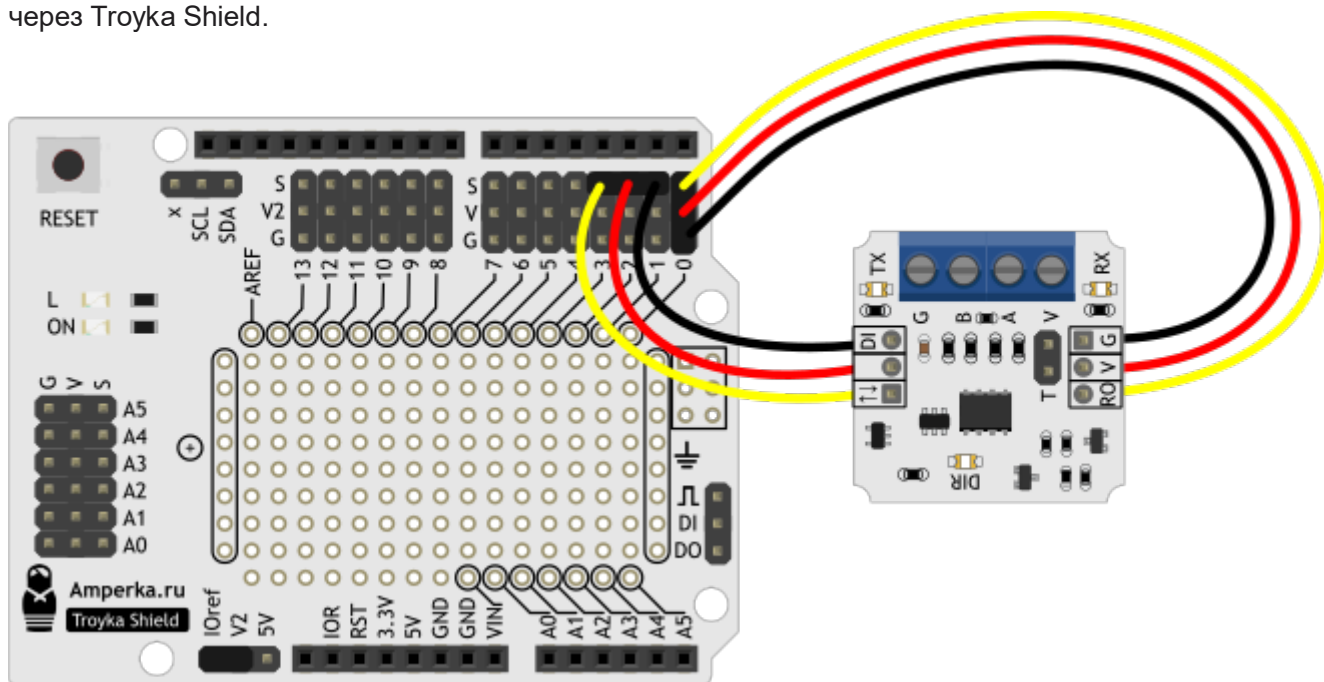
Если хотите оставить минимум проводов — возьмите Troyka Slot Shield.



## SoftwareSerial

Некоторые платы Arduino прошиваются через пины 0 и 1 (Uno, Mega 2560, ADK и Iskra Mini). Перед прошивкой таких плат отключите модуль от пинов RX и TX. Если необходимо одновременно работать с RS485 и подключать контроллер к компьютеру, подключите пины TX и RX к другим контактам управляющей платы.

Для примера подключим управляющие пины RS485-модуля RO и DI — на 8 и 9 пин Arduino через Troyka Shield.

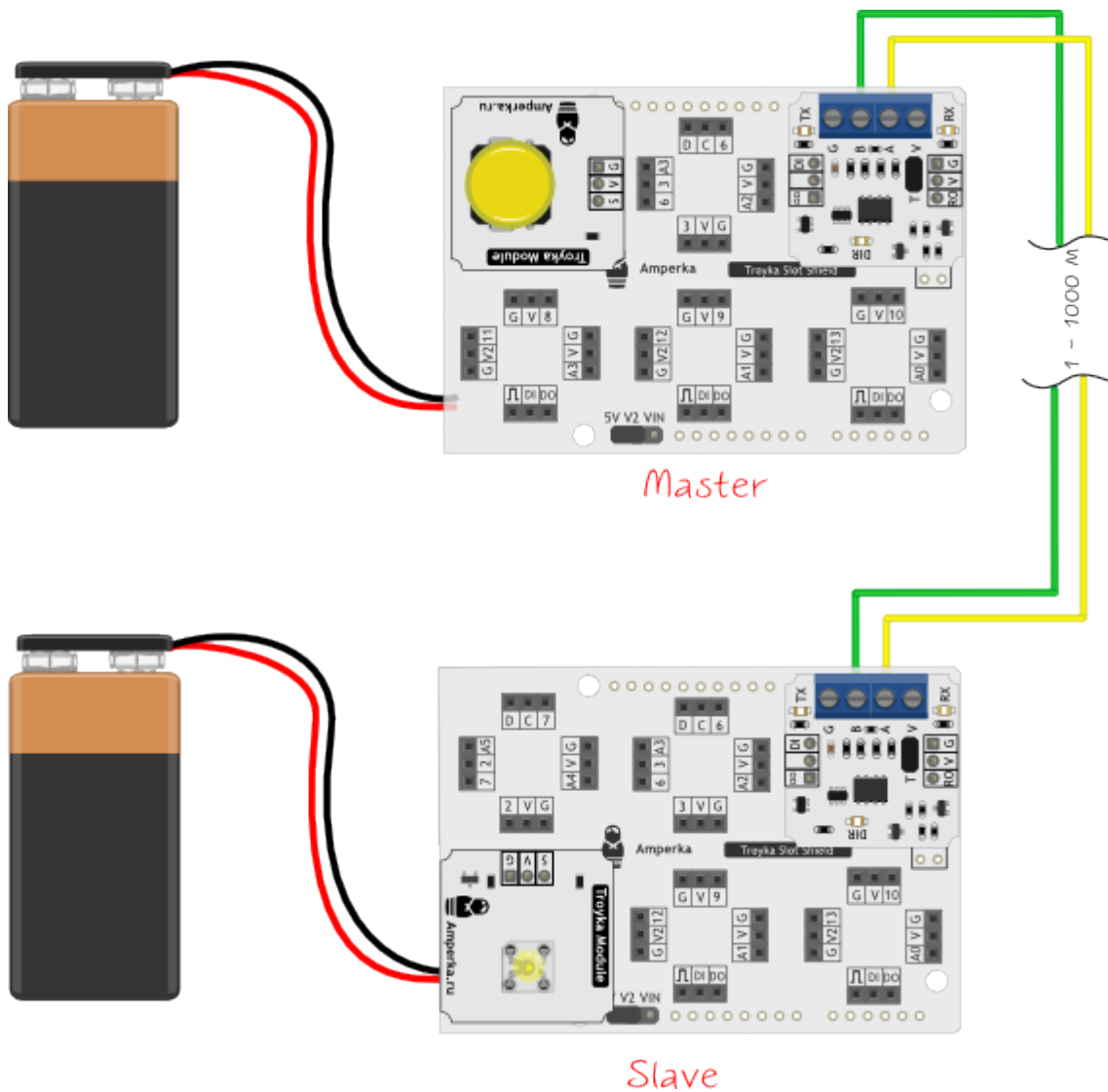


## Примеры работы

RS-485 (Тройка-модуль) работает в полудуплексном режиме — приём и передача идут по одной паре проводов с разделением по времени — один говорит, остальные слушают. Передатчиком может выступать любое устройство в сети. Для этого подайте логическую единицу на пин  $\uparrow\downarrow$  — модуль перейдет в режим передачи. Рассмотрим несколько примеров построения сети на модулях RS-485.

### Простая сеть

Соедините в сеть через RS-485 по двум проводам А и В два устройства Arduino через Тройка Slot Shield. Одно будет передатчиком (*Master*), другое — приёмником (*Slave*). Подключите к *Master* кнопку (Тройка-модуль) к пину А4, а к *Slave* светодиод «Пирания» (Тройка-модуль) к 8 пину.



Прошейте устройства скетчами представленными ниже.

### ***Ведущее устройство***

[masterSimple.ino](#)

```

// даём разумное имя для пинов, к которым подключены кнопки
#define  BUTON_PIN_YELLOW  A4

// была ли жёлтая кнопка отпущена?
boolean buttonStateYellow = true;

// состояние жёлтого светодиода
boolean ledStateYellow = false;

void setup()
{
  // открываем последовательный порт на пинах 1 и 0
  // к которому подключён модуль RS-485
  Serial1.begin(9600);
  // 5 пин в режим выхода
  pinMode(5, OUTPUT);

```

```

    // подаём высокий уровень на 5 пин
    // значит устройство будет передавать данные
    digitalWrite(5, HIGH);
}

void loop()
{
    // отпущена ли кнопка прямо сейчас...
    boolean buttonStateNowYellow = digitalRead(BUTTON_PIN_YELLOW);
    // если кнопка была отпущена и не отпущена сейчас
    if (buttonStateYellow && !buttonStateNowYellow) {
        delay(10);
        // считываем сигнал снова
        buttonStateNowYellow = digitalRead(BUTTON_PIN_YELLOW);
        if (!buttonStateNowYellow) {
            // если она всё ещё нажата
            // значит это клик! Инвертируем сигнал светодиода
            ledStateYellow = !ledStateYellow;
            if (ledStateYellow) {
                // посылаем на модули RS-485 символ «a»
                Serial1.print('a');
            } else {
                // посылаем на модули RS-485 символ «b»
                Serial1.print('b');
            }
        }
    }
    // запоминаем последнее состояние кнопки для новой итерации
    buttonStateYellow = buttonStateNowYellow;
}

```

## ***Ведомое устройство***

[slaveSimple.ino](#)

```

// даём разумное имя для пина, к которому подключен светодиод
#define LED_PIN_YELLOW 8

void setup()
{
    // открываем последовательный порт на пинах 1 и 0
    // к которому подключён модуль RS-485
    Serial1.begin(9600);
    // 5 пин в режим выхода
    pinMode(LED_PIN_YELLOW, OUTPUT);
}

void loop()
{
    // если появились данные с модуля RS-485
    if (Serial1.available() > 0) {

```

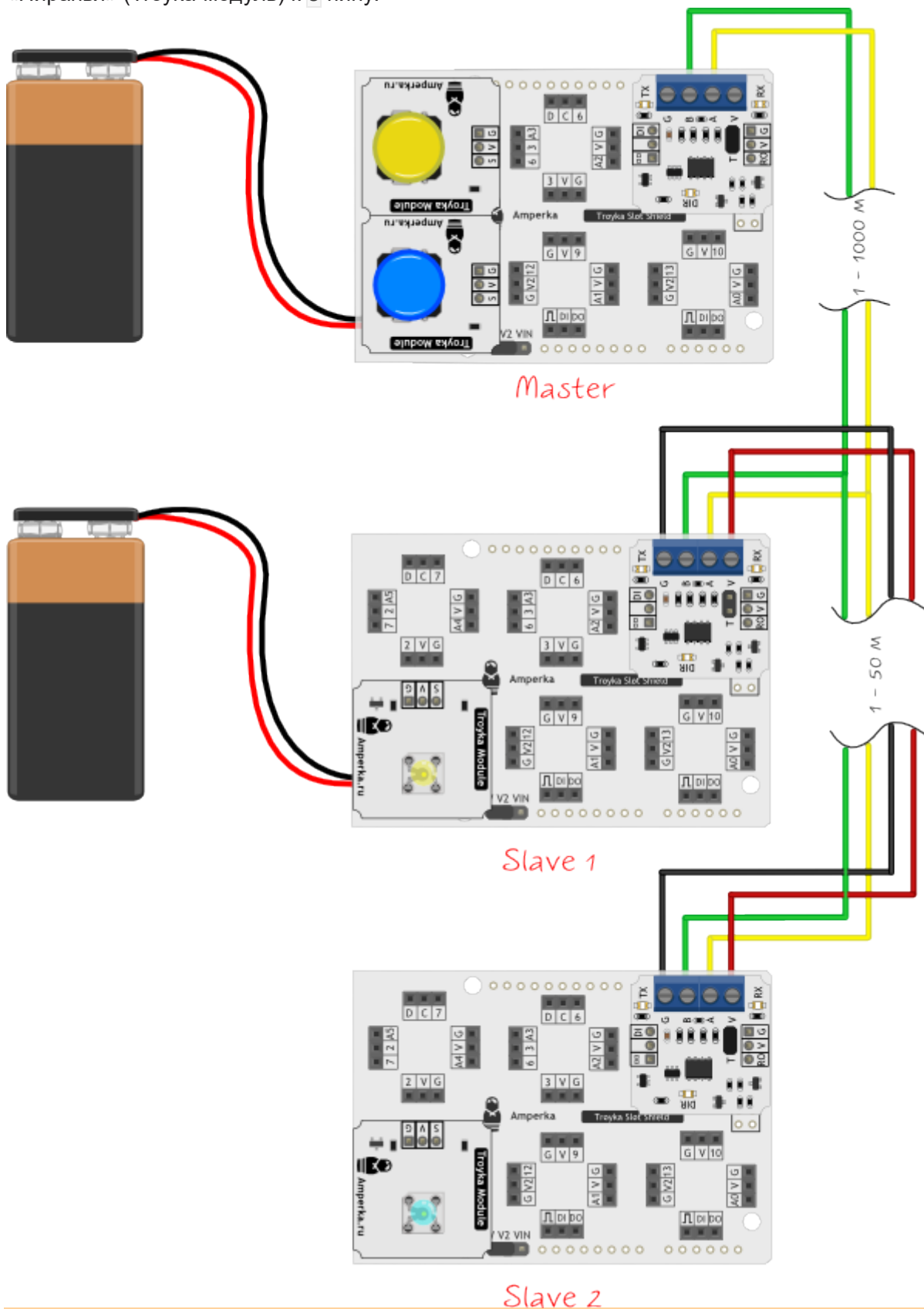
```
// считываем
char c = Serial1.read();
// если символ равен 'a'
if (c == 'a') {
    // зажигаем светодиод
    digitalWrite(LED_PIN_YELLOW, HIGH);
} else if (c == 'b') {
    // если символ равен 'b'
    // гасим светодиод
    digitalWrite(LED_PIN_YELLOW, LOW);
}
}
```

При нажатии на кнопку — светодиод загорится, при повторном — погаснет.

### Сложная сеть

Добавьте в сеть ещё одно устройство *Slave 2*. Соедините контакты питания **V** и **G** у *Slave 1* и *Slave 2*. Так устройство *Slave 2* будет получать не только данные, но и питание от *Slave 1*. Подключите к *Master* ещё одну кнопку (Тройка-модуль) к пину **A3**, а к *Slave 2* светодиод

«Пиранья» (Тройка-модуль) к 8 пину.



При передаче питания по сети длина провода между источником питания и приёмником не должна превышать 50 метров.

Прошейте устройства этими скетчами.

## Ведущее устройство

masterComplex.ino

```
// даём разумное имя для пинов, к которым подключены кнопки
#define  BUTTON_PIN_YELLOW  A4
#define  BUTTON_PIN_BLUE    A3

// была ли жёлтая кнопка отпущена?
boolean buttonStateYellow = true;
// была ли синяя кнопка отпущена?
boolean buttonStateBlue = true;

// состояние жёлтого светодиода
boolean ledStateYellow = false;
// состояние синего светодиода
boolean ledStateBlue = false;

void setup()
{
  // открываем последовательный порт на Serial
  Serial.begin(9600);
  // открываем последовательный порт на пинах 1 и 0
  Serial1.begin(9600);
  // 5 пин в режим выхода
  pinMode(5, OUTPUT);
  // подаём высокий уровень на 5 пин
  // значит устройство будет передавать данные
  digitalWrite(5, HIGH);
}

void loop()
{
  // отпущена ли кнопка прямо сейчас
  boolean buttonStateNowYellow = digitalRead(BUTTON_PIN_YELLOW);
  boolean buttonStateNowBlue = digitalRead(BUTTON_PIN_BLUE);
  // если кнопка была отпущена и не отпущена сейчас
  if (buttonStateYellow && !buttonStateNowYellow) {
    delay(10);
    // считываем сигнал снова
    buttonStateNowYellow = digitalRead(BUTTON_PIN_YELLOW);
    if (!buttonStateYellow) {
      // если она всё ещё нажата
      // значит это клик! Инвертируем сигнал светодиода
      ledStateYellow = !ledStateYellow;
      if (ledStateYellow) {
        // посылаем на модули RS-485 символ «a»
        Serial1.print('a');
      } else {
        // посылаем на модули RS-485 символ «b»
        Serial1.print('b');
      }
    }
  }
}
```



```

    }
}
}
// запоминаем последнее состояние кнопки для новой итерации
buttonStateYellow = buttonStateNowYellow;
// если кнопка была отпущена и не отпущена сейчас
if (buttonStateBlue && !buttonStateNowBlue) {
    delay(10);
    // считываем сигнал снова
    buttonStateNowBlue = digitalRead(BUTTON_PIN_BLUE);
    if (!buttonStateNowBlue) {
        // если она всё ещё нажата
        // значит это клик! Инвертируем сигнал светодиода
        ledStateBlue = !ledStateBlue;
        if (ledStateBlue) {
            // посылаем на модули RS-485 символ «y»
            Serial1.print('y');
        } else {
            // посылаем на модули RS-485 символ «z»
            Serial1.print('z');
        }
    }
}
// запоминаем последнее состояние кнопки для новой итерации
buttonStateBlue = buttonStateNowBlue;
}

```

## **Ведомое устройство 1**

[slaveComplexOne.ino](#)

```

// даём разумное имя для пина, к которому подключен светодиод
#define LED_PIN_YELLOW 8

void setup()
{
    // открываем последовательный порт на пинах 1 и 0
    // к которому подключён модуль RS-485
    Serial1.begin(9600);
    // 5 пин в режим выхода
    pinMode(LED_PIN_YELLOW, OUTPUT);
}

void loop()
{
    // если появились данные с модуля RS-485
    if (Serial1.available() > 0) {
        // считываем
        char c = Serial1.read();
        // если символ равен 'a'
        if (c == 'a') {

```

```

    // зажигаем светодиод
    digitalWrite(LED_PIN_YELLOW, HIGH);
} else if (c == 'b') {
    // если символ равен 'b'
    // гасим светодиод
    digitalWrite(LED_PIN_YELLOW, LOW);
}
}
}

```

## ***Ведомое устройство 2***

slaveComplexTwo.ino

```

// даём разумное имя для пина, к которому подключен светодиод
#define LED_PIN_BLUE 8

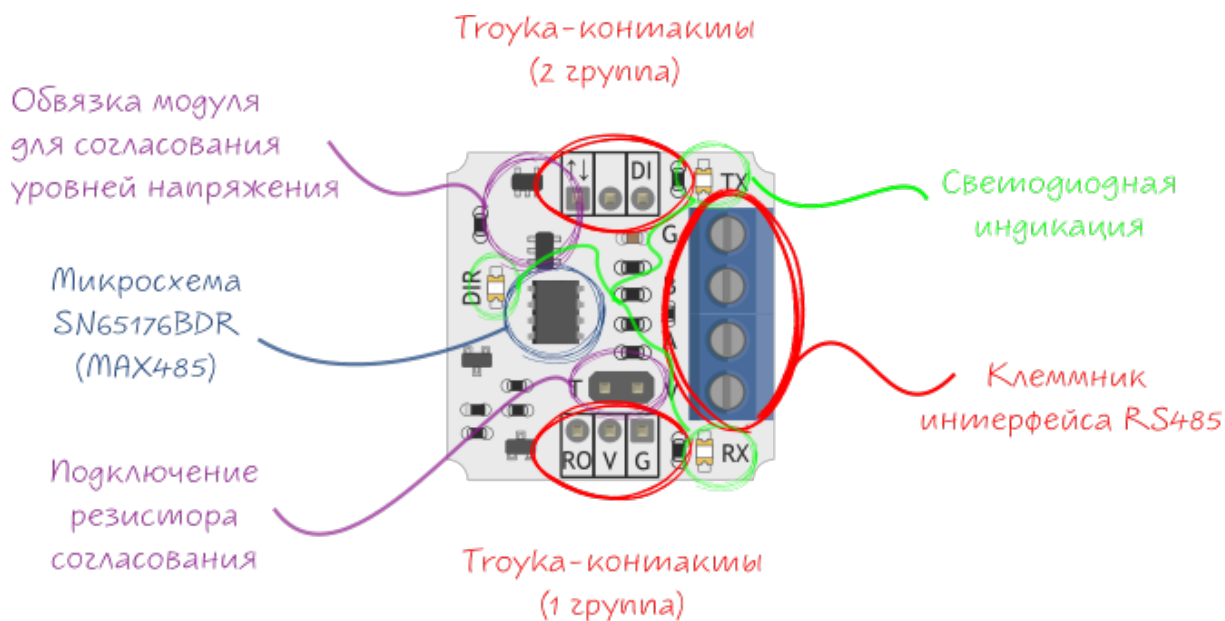
void setup()
{
    // открываем последовательный порт на пинах 1 и 0
    // к которому подключён модуль RS-485
    Serial1.begin(9600);
    // 5 пин в режим выхода
    pinMode(LED_PIN_BLUE, OUTPUT);
}

void loop()
{
    // если появились данные с модуля RS-485
    if (Serial1.available() > 0) {
        // считываем
        char c = Serial1.read();
        // если символ равен 'y'
        if (c == 'y') {
            // зажигаем светодиод
            digitalWrite(LED_PIN_BLUE, HIGH);
        } else if (c == 'z') {
            // если символ равен 'z'
            // гасим светодиод
            digitalWrite(LED_PIN_BLUE, LOW);
        }
    }
}
}

```

При нажатии на жёлтую кнопку — светодиод загорится на *Slave 1*, при повторном — погаснет. Синяя кнопка аналогично включает и выключает светодиод на *Slave2*.

## **Элементы платы**



### Микросхема SN65176BDR

Микросхема SN65176BDR (MAX485) — это приёмопередатчик на интерфейсе RS-485.

RS-485 — один из наиболее распространенных способов передачи сигнала по проводам на длинные расстояния.

Устройства в такой сети соединяются витой парой — двумя скрученными проводами. В основе интерфейса RS-485 лежит принцип дифференциальной (балансной) передачи данных — один сигнала передаётся сразу по двум проводам. По одному проводу (условно А) идет оригинальный сигнал, по другому (условно В) — его инверсная копия. Если на одном проводе  $1$ , то на другом  $0$  и наоборот. Таким образом, между двумя проводами витой пары всегда есть разность потенциалов: при  $1$  она положительна, при  $0$  — отрицательна.

### Клеммник интерфейса RS-485

Клеммники под винт предназначены для подключения устройства в шину RS-485 с возможностью передачи питания.

Обозначения на клеммнике	Назначение
А и В	Для подключения дифференциальной пары проводов (шина RS-485)
V и G	Для передачи или приёма питания от других модулей

### Тройка контакты

#### 1 группа

- Земля (G) — соедините с землёй микроконтроллера.
- Питание (V) — соедините с питанием микроконтроллера.

- Сигнальный (RO) — цифровой выход приемопередатчика. Используется для передачи данных в микроконтроллер. Подключите к пину **RX** микроконтроллера.

#### 2 группа

- Сигнальный (↑↓) — подключите к сигнальному пину микроконтроллера. Служит для выбора режима модуля — приёмник или передатчик. При высоком уровне — работа на прием, при низком — на передачу.
- Не используется.
- Сигнальный (DI) — цифровой вход приемопередатчика. Используется для приёма данных из микроконтроллера. Подключите к пину **TX** микроконтроллера.

#### Светодиодная индикация

Имя светодиода	Назначение
RX и TX	Мигают при обмене данными между RS485 и управляющим устройством.
DIR	Горит, если модуль в режиме передатчика

#### Обвязка для согласования уровней напряжения

Необходима для сопряжения устройств с разными питающими напряжениями. В нашем случае это может быть управляющая плата Iskra JS с **3,3** вольтовой логикой и RS485(Тройка-модуль) с **5**вольтовой логикой.

#### Характеристики

- Напряжение питания: 3,3–5 В
- Потребляемый ток: 10 мА
- Максимальное расстояние связи: 1200 м
- Максимальное количество узлов в сети: 32
- Интерфейс: последовательный порт (UART)
- Чип RS-485: SN65176BDR (MAX485)
- Габариты: 25,4×25,4 мм