

МИНИСТЕРСТВО ТРУДА, ЗАНЯТОСТИ И ТРУДОВЫХ РЕСУРСОВ
НОВОСИБИРСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ НОВОСИБИРСКОЙ ОБЛАСТИ
«НОВОСИБИРСКИЙ КОЛЛЕДЖ ЭЛЕКТРОНИКИ И
ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ»

РЕСУРСНЫЙ ЦЕНТР МИКРОЭЛЕКТРОНИКИ И
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

МЕТОДИЧЕСКАЯ ПУБЛИКАЦИЯ

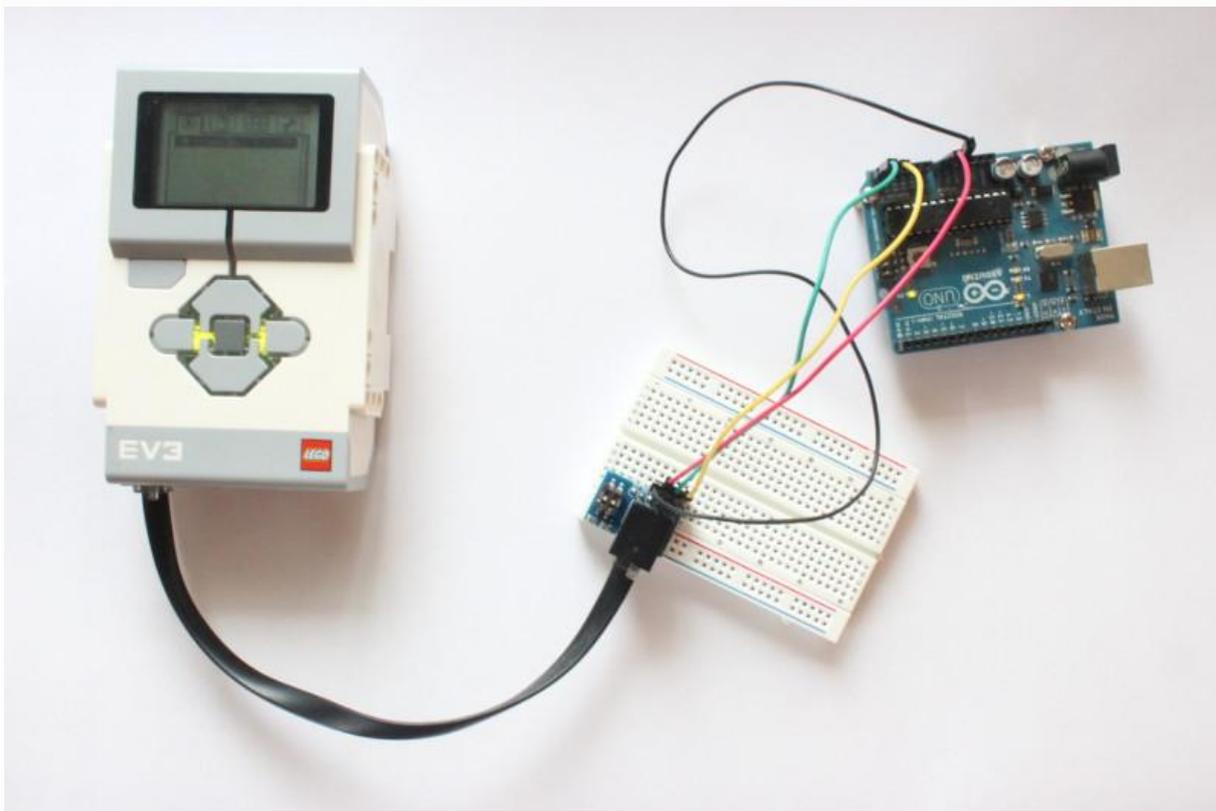
«Соединение LegoEV3 и Arduino»

Методист:  Г.В. Комарова

Новосибирск 2016 г.

Иногда, при создании роботов возникает необходимость использования нескольких различных платформ в одном проекте, например, EV3 и Arduino. Так вы сможете использовать сильные стороны каждой из платформ и распараллелить выполняемые задачи. Предлагаем вам соединить EV3 и Arduino и как использовать Arduino совместно с EV3.

В примере мы используем макетную плату. Вы можете использовать адаптер для подключения NXT к макетной плате, чтобы не портить NXT-кабель из вашего набора.



Также продемонстрируем, как использовать Arduino для чтения аналоговых значений и отправки их на EV3. Отправка и получения любых данных на EV3, начиная от показаний акселерометра и закидывая значениями снятых со счётчика Гейгера.

Общение EV3 и Arduino может быть организовано через I2C. В этом руководстве мы настроим Arduino как I2C-слейв (т.е. подчинённое устройство), а EV3 – как I2C-мастер (т.е. главное устройство).

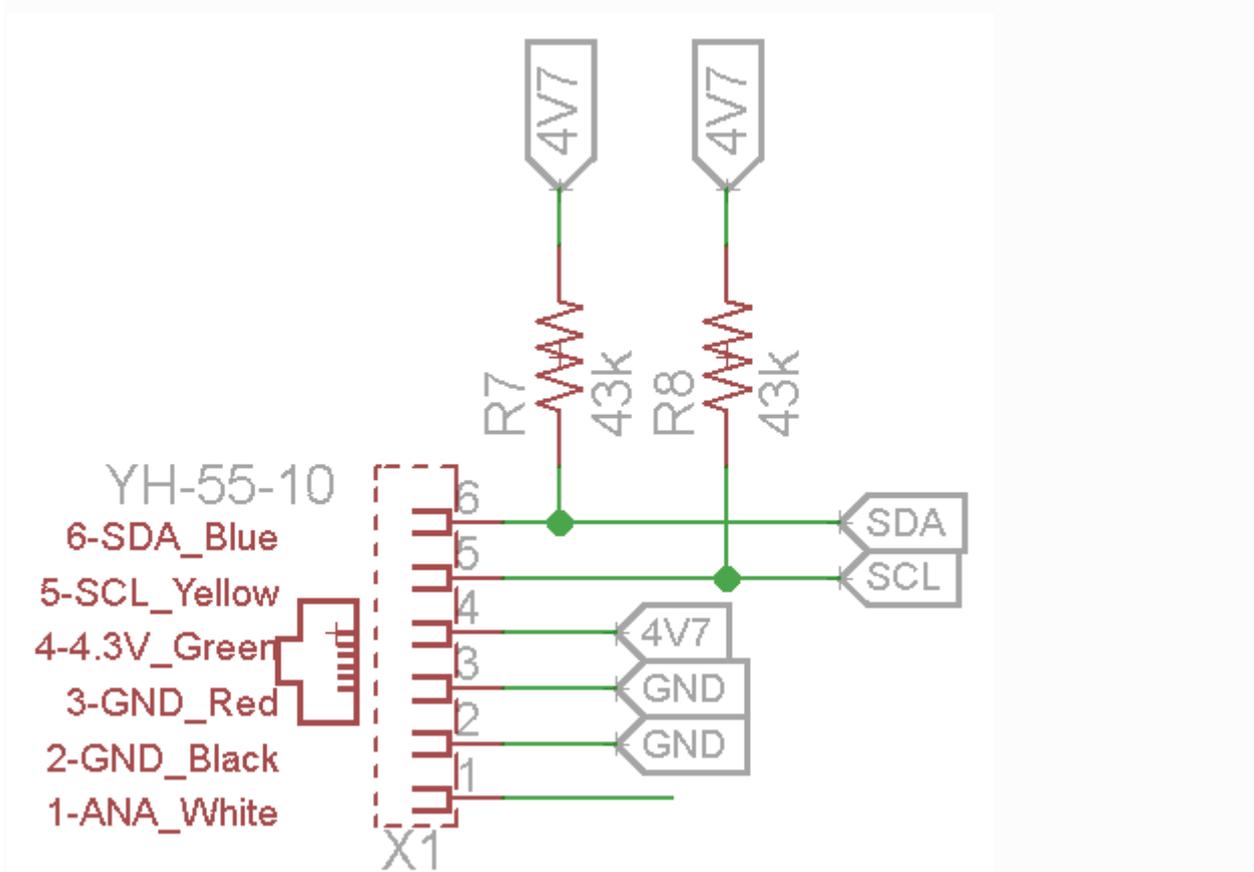
В примере мы покажем вам, как отправлять команды на Arduino из EV3, и как написать программу, которая будет запрашивать данные из Arduino.

Подготовка

Шину I2C называют «интерфейсом двух проводов», потому что она использует для связи два провода. Один провод передаёт такты: это гарантирует, что оба устройства посылают информацию с одинаковой

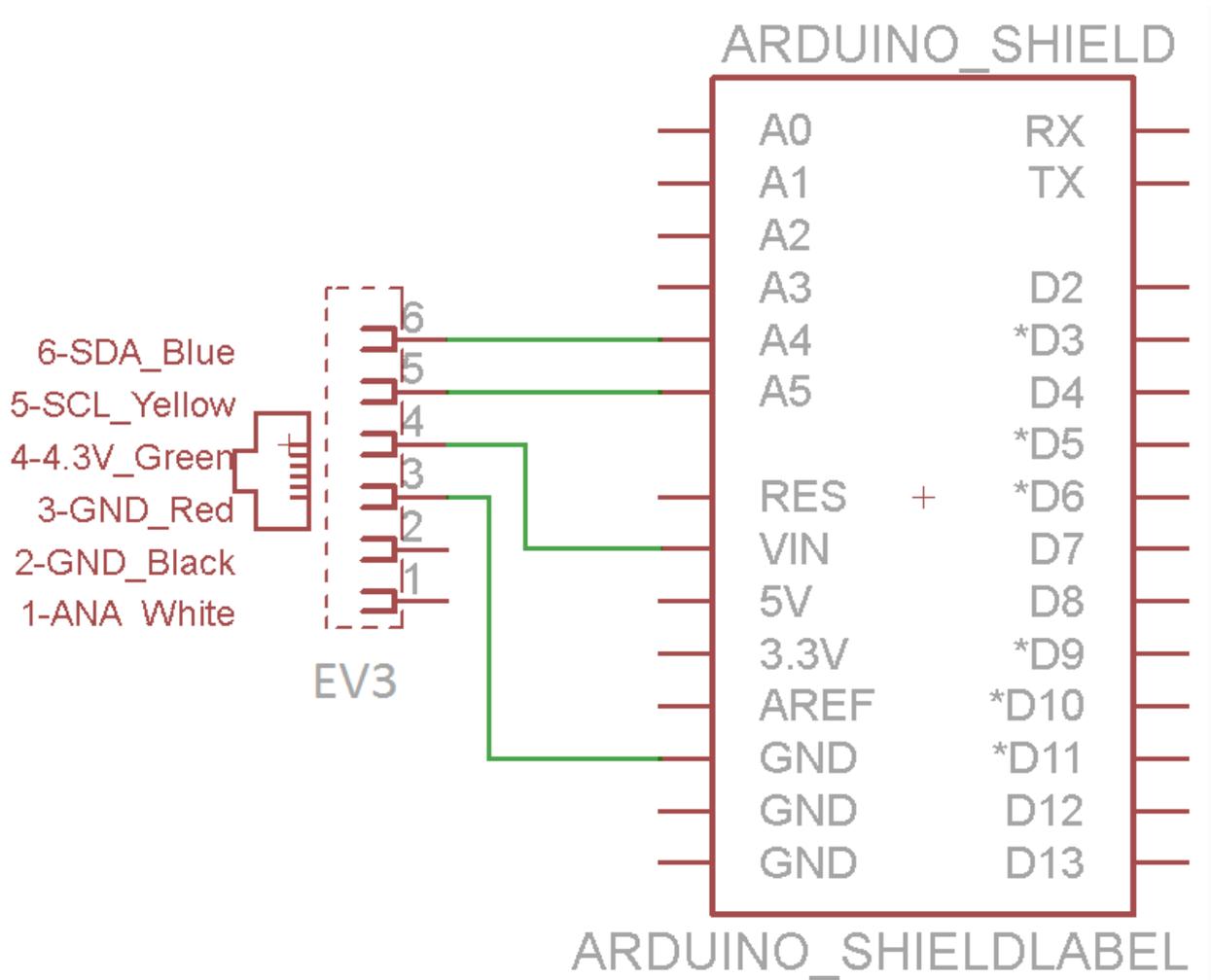
скоростью. Линию тактов обычно называют «SCL» (сокращённо от SerialClock). Второй провод служит для передачи данных (данные отправляются и принимаются обоими устройствами одновременно), и обычно называется «SDA» (сокращённо от SerialData). Мастер (всегда EV3) всегда контролирует соединение, и всегда устанавливает такты для обоих устройств. Контролируемое устройство обычно называется слейв (Slave – раб или подчинённый по англ.). Когда мастер посылает команды, слейв бросает всё и слушает. Когда мастер требует информацию, слейв отдаёт эту информацию.

В примере мы не используем резисторы подтяжки на I2C линиях. Однако, если I2C совсем не работает, вы можете использовать 47кОм резисторы подтяжки на SDA и SCL линиях, чтобы подтянуть I2C до 4,7В на линии VCC.

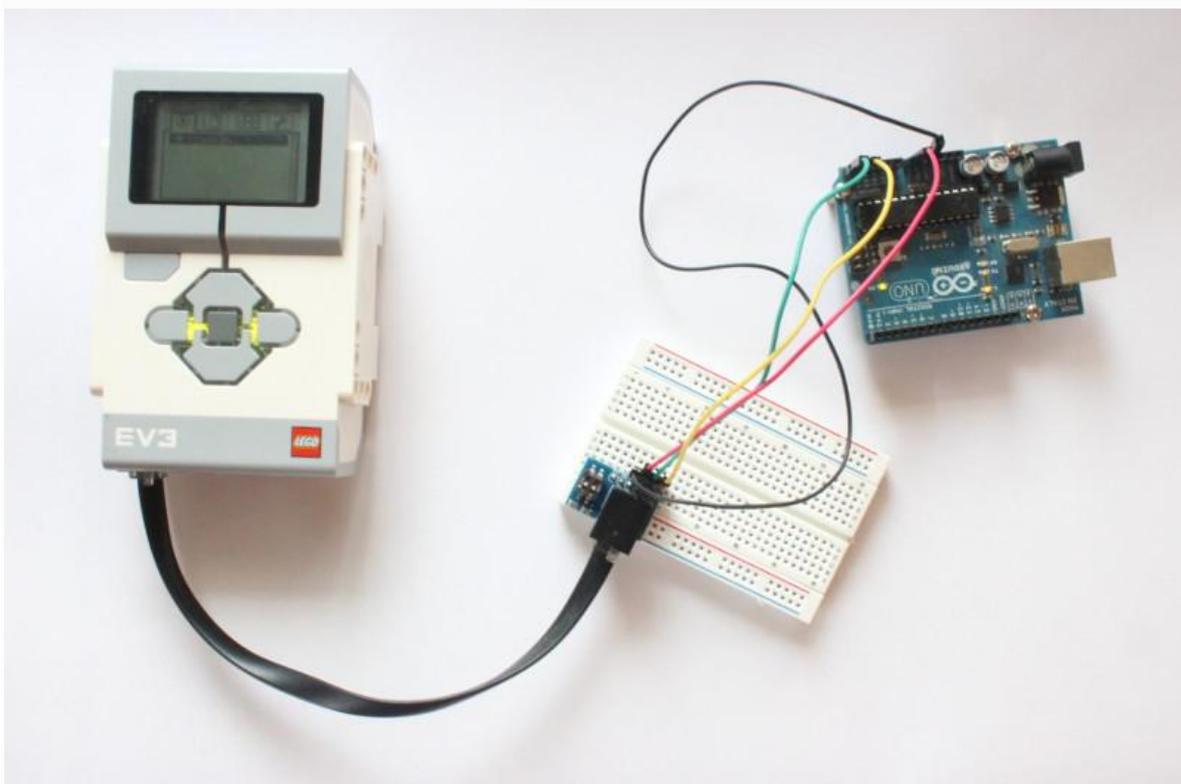


Настройка оборудования

На левой стороне схемы показана вилка EV3, а на правой стороне – Arduino. Мы должны соединить GND, VCC, SDA и SCL.



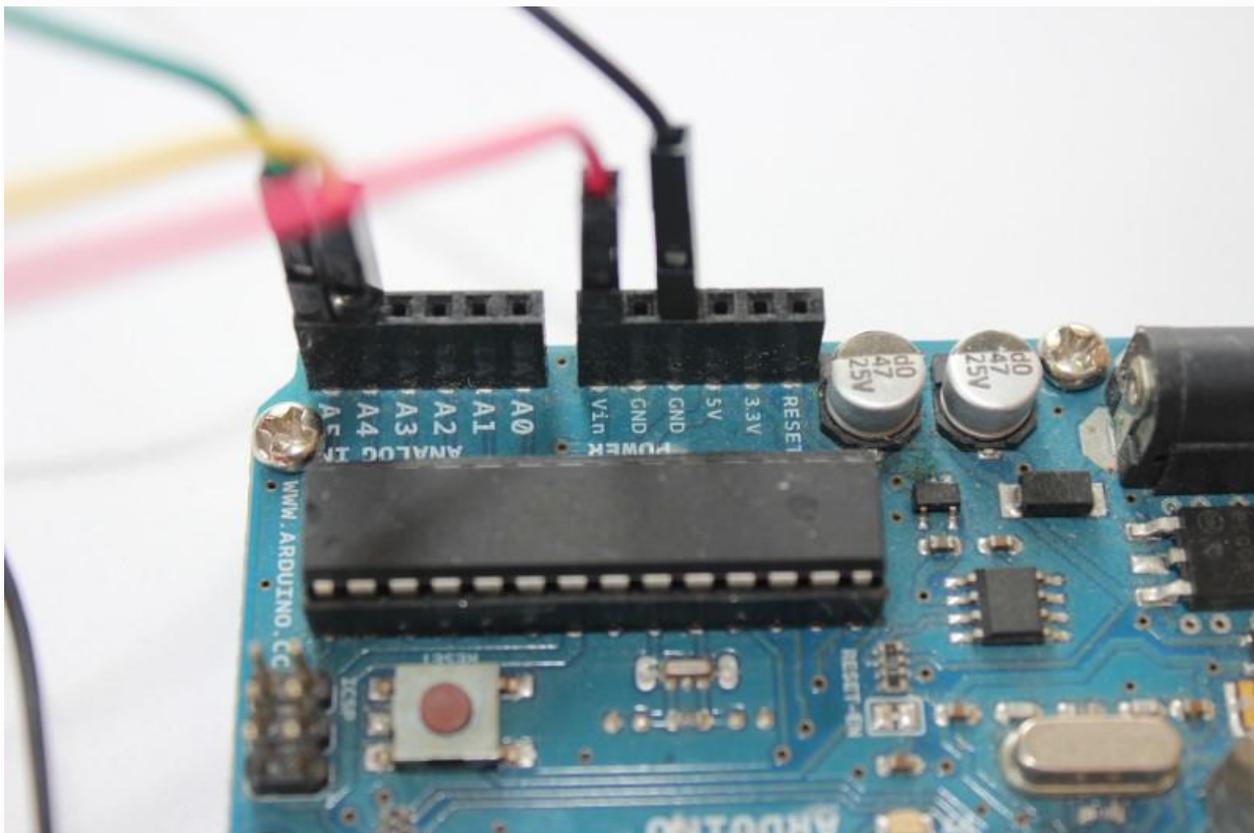
Соединённые провода будут выглядеть так: **EV3 -> макетная плата -> Arduino**. Внизу на картинке показано как выглядит это соединение.



В примере мы подключим только линии SDA и SCL (для обмена данными), VCC для питания Arduino от EV3 и линию земли (GND) для выравнивания напряжения.

Подключение к Arduino

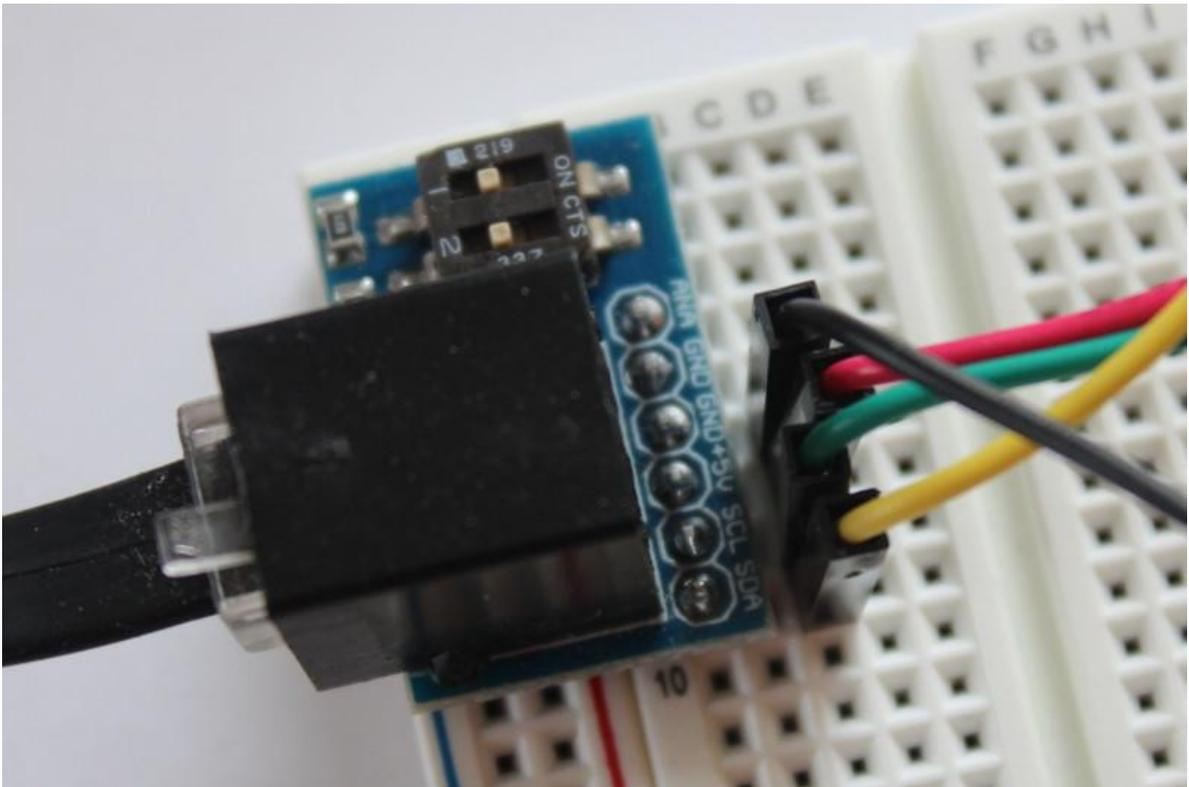
Сперва подключите провод к порту GND Arduino. Затем подключите один провод к A5 (SCL, линия тактов) и один к порту A4 (SDA, линия данных). На картинке вы можете видеть, что мы подключили красный провод к VCC, чёрный – к GND, жёлтый – к SDA (A4) и зелёный - к SCL (A5).



Подключение к EV3

Наиболее простой способ подключить EV3 – это использовать адаптер. Все цвета идут в соответствующие пины: жёлтый провод в примере подключен к SDA на макетной плате. Тоже для SCL, GND и VCC.

Для наглядности: SDA → SDA, SCL → SCL.



На этой картинке цвета соответствуют следующим линиям:

- Чёрный: земля
- Красный: VCC
- Зелёный: SCL
- Жёлтый: SDA

На стороне Arduino мы подключим SDA, SCL, VCC и землю в разъёмы Arduino.

Программное обеспечение

Программное обеспечение для сопряжения EV3 и Arduino состоит из двух частей. Первая часть – это скетч, выполняемый на Arduino, вторая часть – программа для среды LEGO MINDSTORMS EV3 с установленными блоками, которые отправляют и принимают данные на/от Arduino.

Настройка Arduino для работы с I2C: Присвоение адреса

Нам нужно выбрать адрес слейва. В примере мы установили адрес слейва 0x04 (в шестнадцатеричной системе счисления).

Arduino как слейв получатель/отправитель

В нашем примере, Arduino будет получать данные от мастера. EV3 просто говорит «вот некоторые данные», а Arduino говорит «спасибо за

данные». Настройка очень простая и большинство датчиков работают как слейв получатели и отправители. Этот пример может быть очень полезен, если вы настраиваете свой Arduino для управления мотором или LED.

EV3 отправляет некоторые данные на Arduino, который получает их и выводит их в SerialMonitor. Если EV3 запрашивает некоторые данные, Arduino отправляет в ответ один байт. Вы можете легко изменить этот код, чтобы отправлять или получать любое количество байт.

```
#include <Wire.h>

#define SLAVE_ADDRESS 0x04

void setup()
{
  Serial.begin(9600); // Инициализируем серийный порт для вывода.
  Wire.begin(SLAVE_ADDRESS);
  Wire.onReceive(receiveData);
  Wire.onRequest(sendData);
  Serial.println("Ready!");
}

int val, flag = 0;

void loop()
{
  if(flag == 1)
  {
    Serial.print(val);
    flag = 0;
  }
}

void receiveData(int byteCount)
{
  while(Wire.available() > 0)
  {
    val = Wire.read();
```

```

flag = 1;
  }
}
// Функция для отправки данных.
void sendData()
{
  Wire.write(0x45);
}

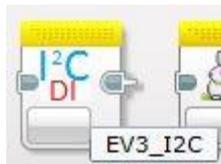
```

Функция **setup()** инициализирует серийный порт и указывает, какие функции будут вызываться при отправке и получении данных от EV3.

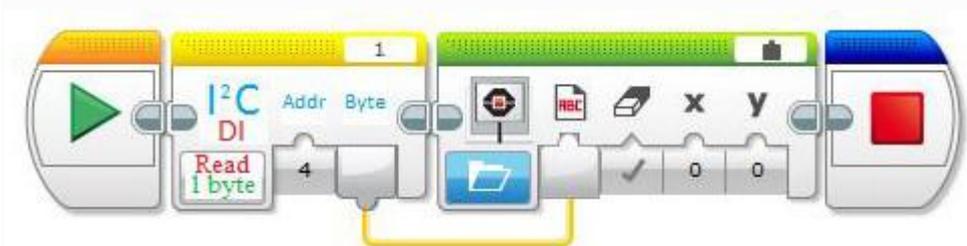
Функция **receiveData()** используется для получения данных от EV3, а **sendData()** – для отправки данных на EV3. Загрузите этот скетч на Arduino, для использования с программным обеспечением EV3.

EV3 как мастер

Настройка EV3 как мастера очень простая и импортируйте их в ПО LegoMindstorms EV3 (Инструменты -> Мастер импорта блоков). Используйте блок **DexterIndustries I2C** для взаимодействия с Arduino.



Для чтения одного байта, просто создайте следующую последовательность блоков с включенным режимом «**Read 1 Byte**». Пропишите в поле «**Addr**» адрес слейва, такой же, как в скетче для Arduino (в нашем случае **0x04**) и запустите программу, чтобы посмотреть данные пришедшие от Arduino на экране EV3.



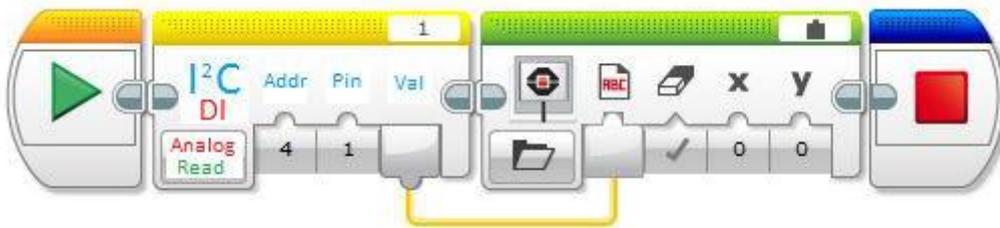
Для отправки одного байта, поменяйте режим блока на «**Write 1 Byte**» и введите байт, который вы хотите отправить на Arduino. Когда вы запустите программу, EV3 пошлёт данные на Arduino и Arduino отобразит их в терминале серийного порта.



Аналогично вы можете загрузить в Arduino скетч для чтения/записи 8-ми байт и использовать в блоке DexterIndustries I2C режим «**8 byteRead/Write**».



Иногда очень полезно снимать аналоговые данные на Arduino и отправлять их на EV3. Если вы хотите делать так, то загрузите этот скетч на Arduino, а в EV3-программе выберите в блоке режим «**AnalogReadBlock**». Затем укажите номер пина, с которого вы хотите считывать данные и запустите программу. После этого на EV3 будут приходить аналоговые данные.



Вот и всё про соединение EV3 и Arduino.