

[КАК СТАТЬ АВТОРОМ](#)[Тест для аналитиков: какой из вас штурман](#)**kubanbanzai**

30 янв 2014 в 21:49

# Управление устройством USB HID на Windows 7

9 мин 91K

C#\*

В статье сделана попытка предоставить пошаговую инструкцию — как соединить самодельное устройство **USB HID** на микроконтроллере **AVR** и компьютер с операционной системой **Windows 7 x64**, чтобы обмениваться данными и управлять портами микроконтроллера. Пример приложения управляет через USB ножкой порта микроконтроллера (к ней подключен индикационный светодиод). Есть возможность также прочитать состояние светодиода — потушен он или горит. Топик предназначен для новичков, поэтому большая просьба к знатокам программирования — приберегите тухлые яйца и гнилые помидоры иронические комментарии для более удобного случая.

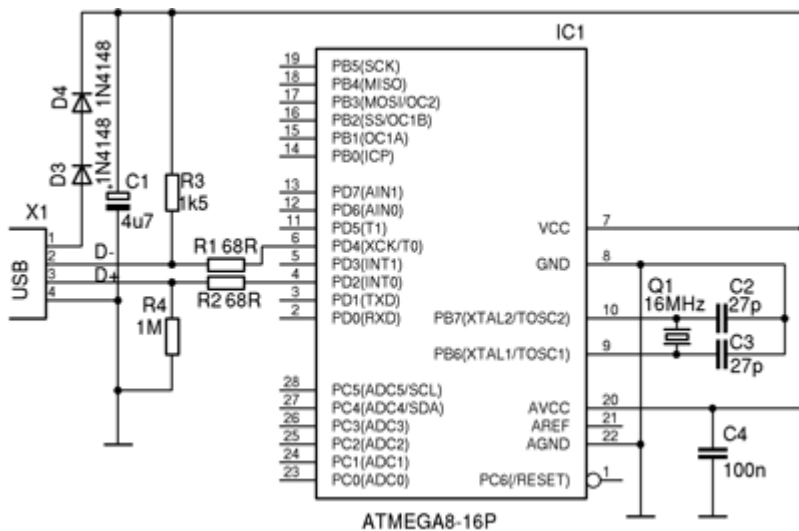
## Используемое программное обеспечение

1. Для микроконтроллера — библиотека V-USB [1] компании Objective Development и IDE Atmel Studio 6 [2] компании Atmel. Нужно также скачать и установить тулчейн WinAVR [3] для компиляции firmware микроконтроллера (для спецов это необязательно, потому что можно обойтись тулчейном, который входит в состав Atmel Studio).
2. Для написания программы Windows (ПО хоста) использовалась библиотека LibUsbDotNet [4] Тревиса Робинсона и IDE Visual Studio C# 2010 [5] компании Microsoft.

Все программное обеспечение, кроме Visual Studio 2010, бесплатное, хотя есть возможность использовать Visual Studio C# 2010 Express бесплатно в течение ограниченного срока. Все действия проводились в среде операционной системы Windows 7 x64, но наверняка подойдет и любая другая операционная система семейства Windows (Windows XP и более свежая).

## Используемое железо

Благодаря библиотеке V-USB для создания устройства USB HID подойдет любой микроконтроллер AVR. Если Вы дружите с паяльником, то даже можете собрать подключение к USB самостоятельно по одной из опубликованных схем. Такая схема (взята из пакета V-USB [1]) в качестве примера приведена на картинке.



Чтобы экономить время и усилия, лучше использовать готовую макетную плату. Особенно удобно, если в плату будет записан USB-загрузчик (bootloader), тогда не понадобится покупать программатор для перепрошивки платы. Я использовал макетную плату AVR-USB-MEGA16 с микроконтроллером ATmega32A, в ней загрузчик есть (USBasploader, эмулирующий поведение программатора USBasp). Вот так платка выглядит в натуральную величину:

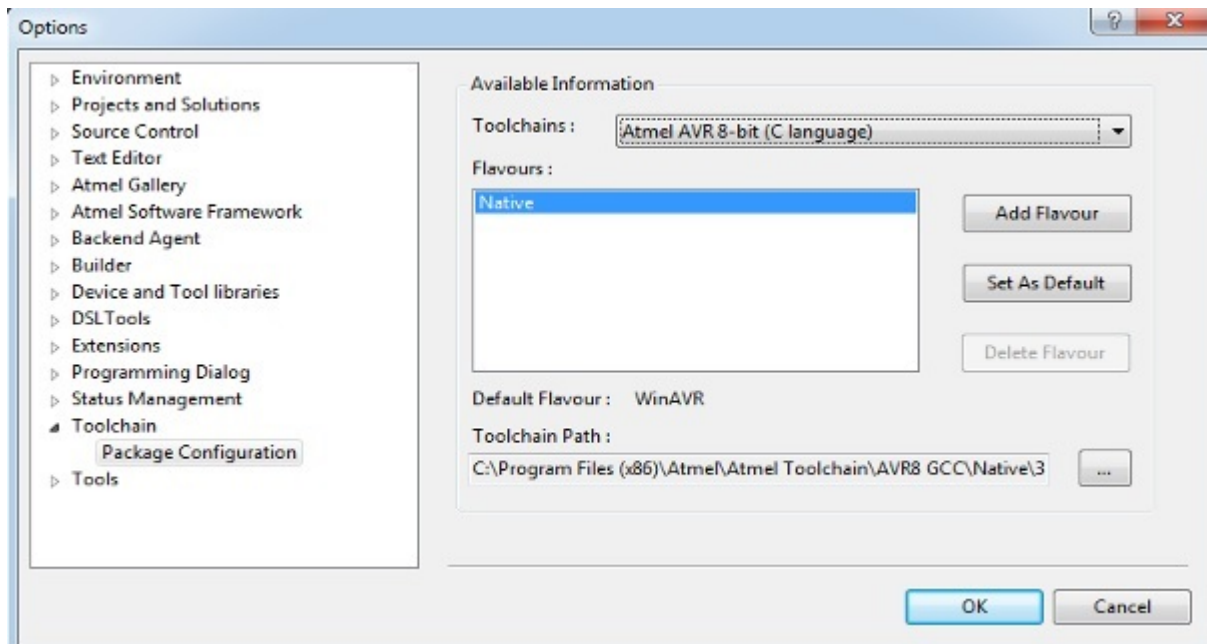


Можно взять также metaboard (на нем стоит ATmega168 или ATmega328), или даже программатор на микроконтроллере ATmega8. Подобные железки можно дешево купить на [ebay.com](https://www.ebay.com) или [dx.com](https://www.dx.com).

### Создание firmware микроконтроллера с помощью Atmel Studio 6 и библиотеки V-USB

Сделайте новый проект в Atmel Studio 6 (далее просто AS6). Когда AS6 предложит выбрать микроконтроллер, выберите **Atmega32** без буквы **A**, не Atmega32A (хотя на плате стоит **Atmega32A**) — это важно, так как тулчейн WinAVR не видит разницы, он знает только Atmega32. Эти микроконтроллеры по внутреннему устройству идентичны, так что для нас разницы нет, а для AS6 есть.

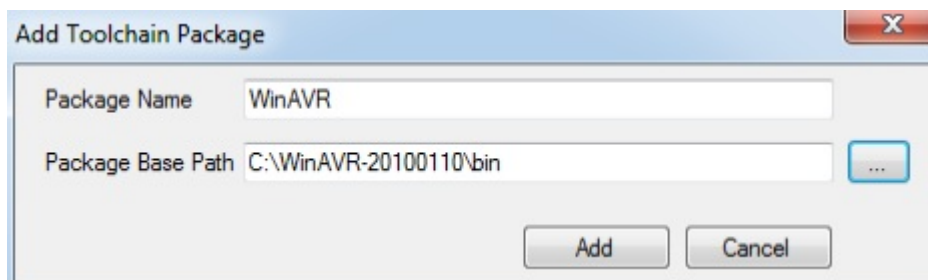
Теперь нужно правильно настроить компилятор. В верхнем меню AS6 нажмите **Tools**, далее **Options..** и появится вот такое окно:



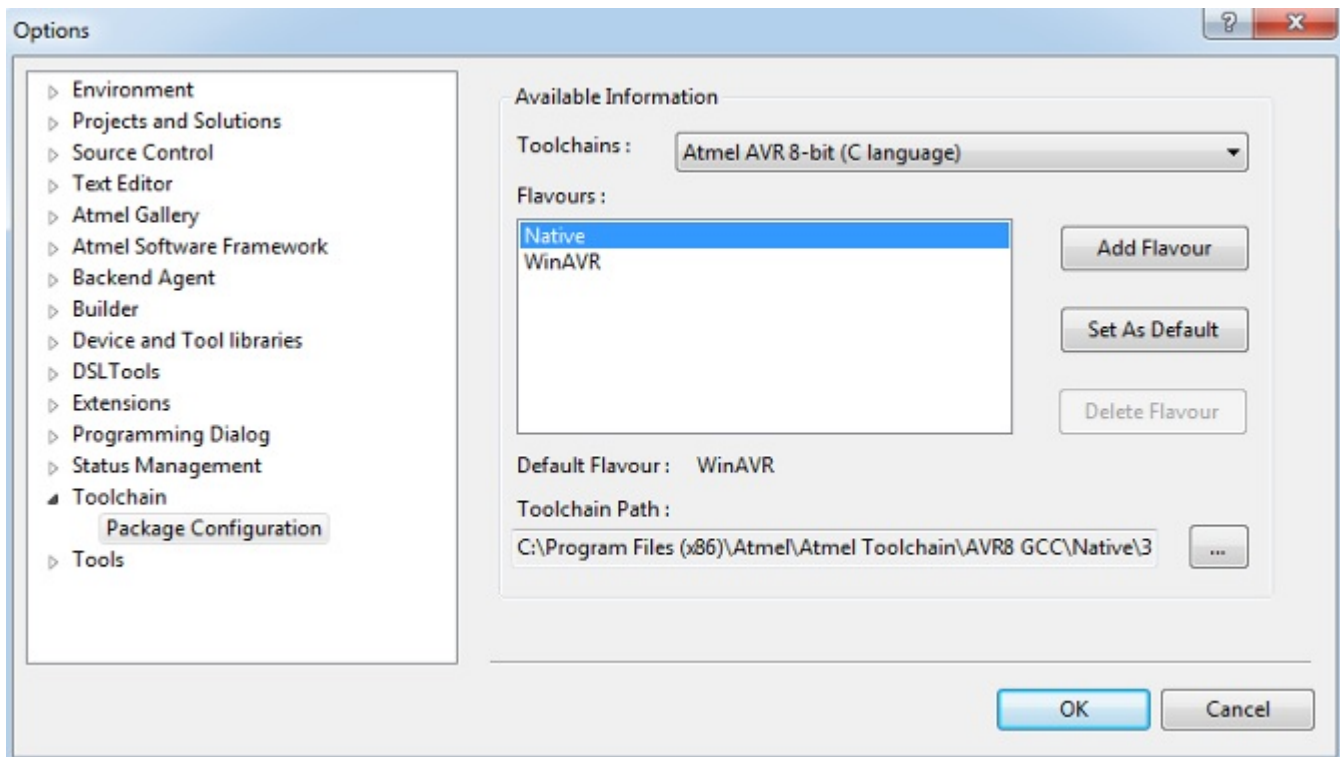
Слева в списке выберите **Toolchain**. Справа появится список Flavours. Этим словечком Atmel закодировала возможные варианты используемого инструментария (тулчейны).

Примечание. В списке уже присутствует тулчейн Native, который используется по умолчанию

Для добавления в список Flavours тулчейна WinAVR нажмите кнопку **Add Flavour**, появится следующее окно:

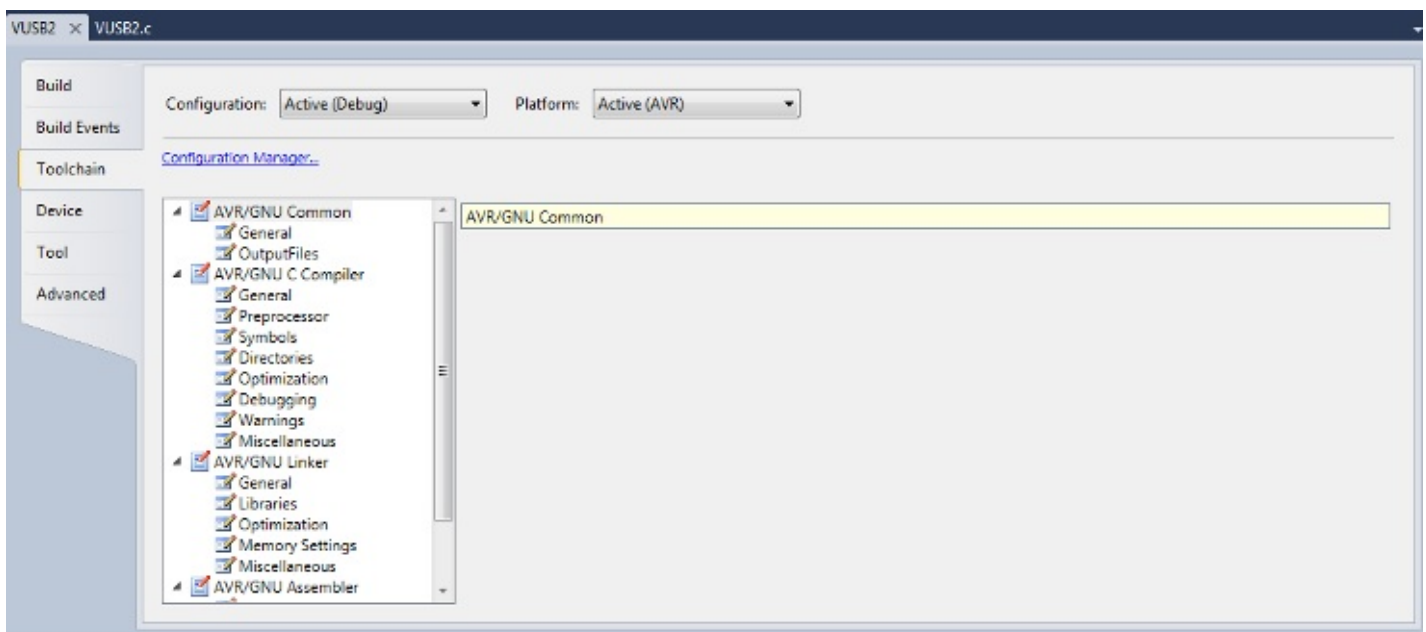


В верхней строчке этого окна введите имя компилятора WinAVR (произвольное), а в нижней строчке введите полный путь, куда установлен сам компилятор тулчейна (с указанием папки \bin) и нажмите кнопку **Add**. В списке Flavours появится добавленный компилятор, как показано на скриншоте.



Выделите мышкой наш новый добавленный компилятор WinAVR и нажмите кнопку **Set As Default** (сделать его тулченом по умолчанию), и нажмите OK. После этой процедуры наша AS6 будет использовать компилятор WinAVR.

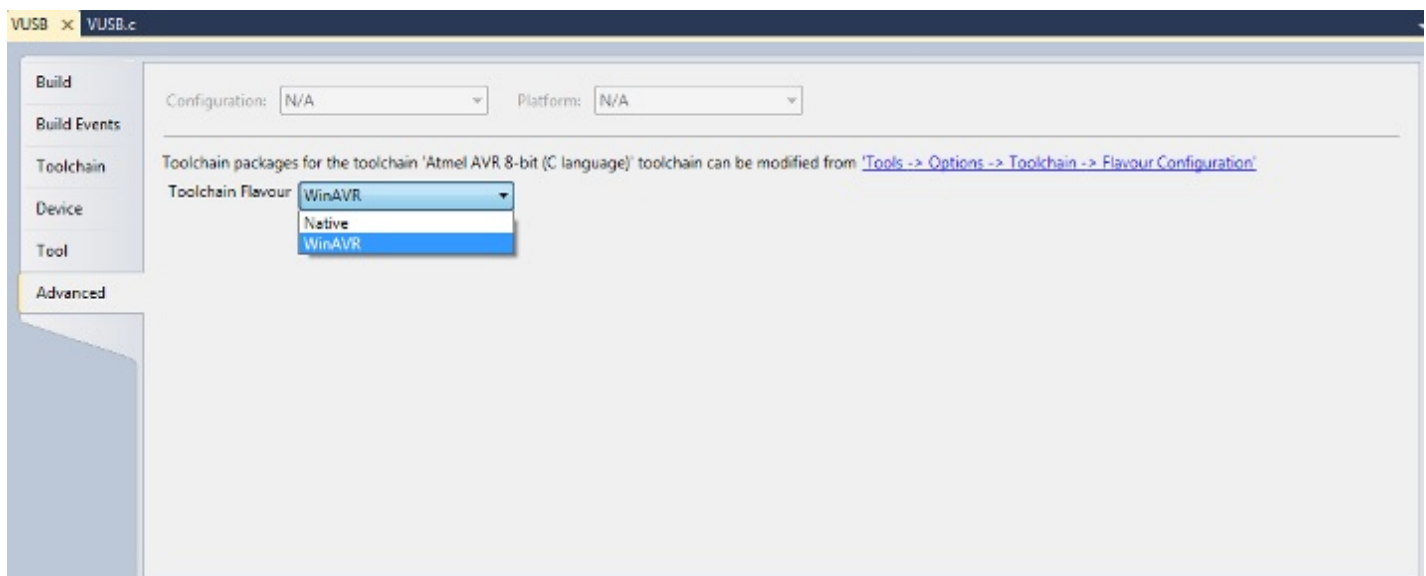
Пора настроить свойства нашего проекта, для этого курсором в Solution Explorer левым щелчком выберите имя проекта и нажмите **Alt+F7** (меню Project -> Properties), появится окно с настройками:



Сделайте следующие настройки:

- В разделе **AVR/GNU C Compiler -> Symbols** добавляем в поле **-D** строку **F\_CPU=12000000UL** — это соответствует частоте микроконтроллера 12 МГц (такой кварц установлен на моей макетной плате AVR-USB-MEGA16).
- В разделе **AVR/GNU Assembler -> General** в поле **Assembler flag** надо добавить **DF\_CPU=12000000UL**.
- В разделе **AVR/GNU C Compiler -> Optimization** в поле **Optimization Level** должно стоять **Optimize for size (-Os)**.

Далее очень важный момент — в левой части окна, в списке выберите раздел **Advanced**, как показано на рисунке ниже.



В выпадающем списке **Toolchain Flavour** выберите добавленный компилятор WinAVR, чтобы при компиляции проекта AS6 использовала его. На этом настройка AS6 закончена.

Далее необходимо в созданный проект добавить файлы исходного кода проекта [6] — см. папку firmware\VUSB, файлы VUSB.c, usbdrv.c, usbdrvasm.S и oddebug.c. Проект ASS6 создан на основе одного из примеров библиотеки V-USB: hid-custom-rq, который изначально компилировался с помощью утилиты make из командной строки. На основе библиотеки V-USB можно найти много других примеров кода — в основном это устройства USB HID (мыши, клавиатуры, устройства ввода и вывода), но есть также и устройства USB CDC (виртуальный COM-порт). Если Вам лень самому создавать проект, просто откройте в AS6 файл проекта VUSB.at9ln, в нем уже сделаны все необходимые настройки и добавлены все нужные файлы.

Если у Вас используется другая макетная плата, то нужно правильно настроить файл *usbconfig.h*. Это конфигурационный файл библиотеки V-USB, в нем задаются многие

настройки и параметры (VID, PID, ножки микроконтроллера, значения для дескрипторов и другие настройки). Подробное описание всех настроек дано в комментариях этого файла. Основное внимание следует уделить назначению выводов микроконтроллера, которые используются под сигналы USD D+ и D- (макроопределения USB\_CFG\_IOPORTNAME, USB\_CFG\_DMINUS\_BIT, USB\_CFG\_DPLUS\_BIT), к этим ножкам предъявляются особые требования. Конфигурационный файл *usbconfig.h* из архива [6] предназначен под разводку ножек макетной платы AVR-USB-MEGA16, и он гарантированно работает. Моргать программа будет светодиодом, который уже имеется на макетной плате и подключен к ножке 0 порта B.

## Создание программы для компьютера (ПО хоста)

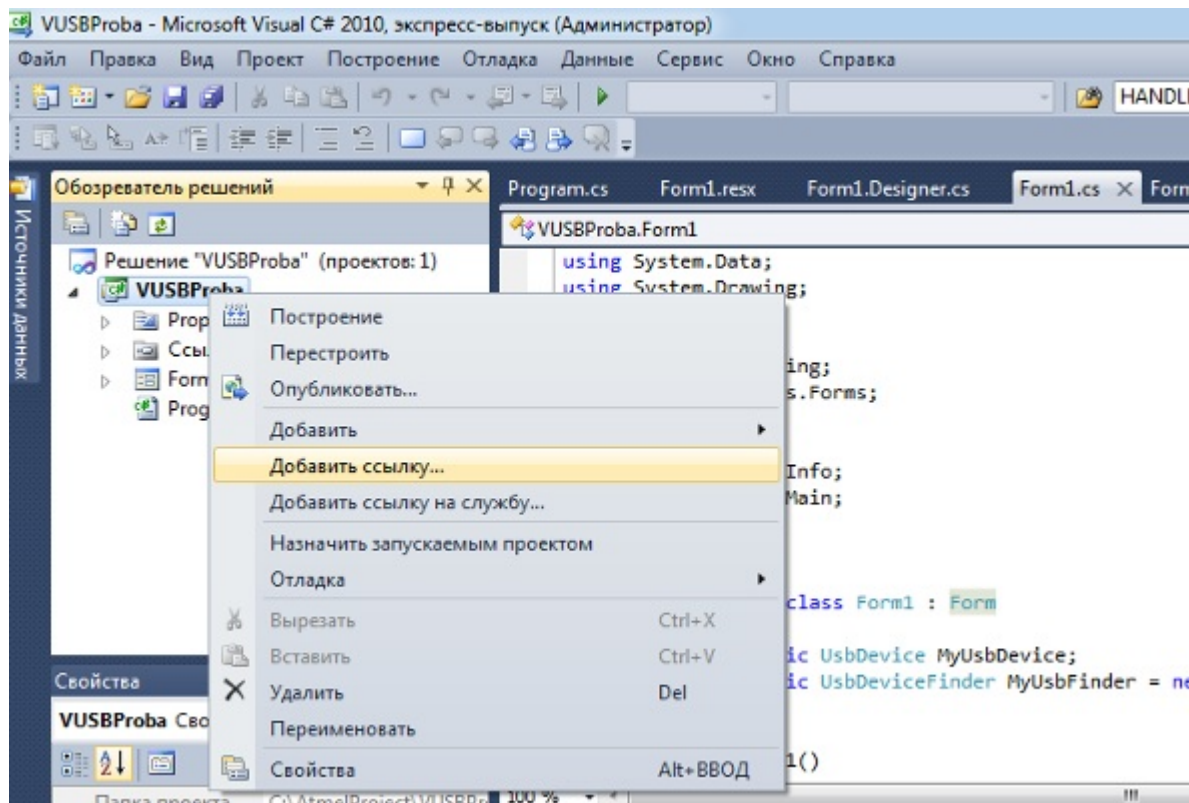
Наша программа должна посылать через подключение USB пакеты, которые будут управлять микроконтроллером.

Примечание. Программа была создана на основе примера консольного приложения из той же 6

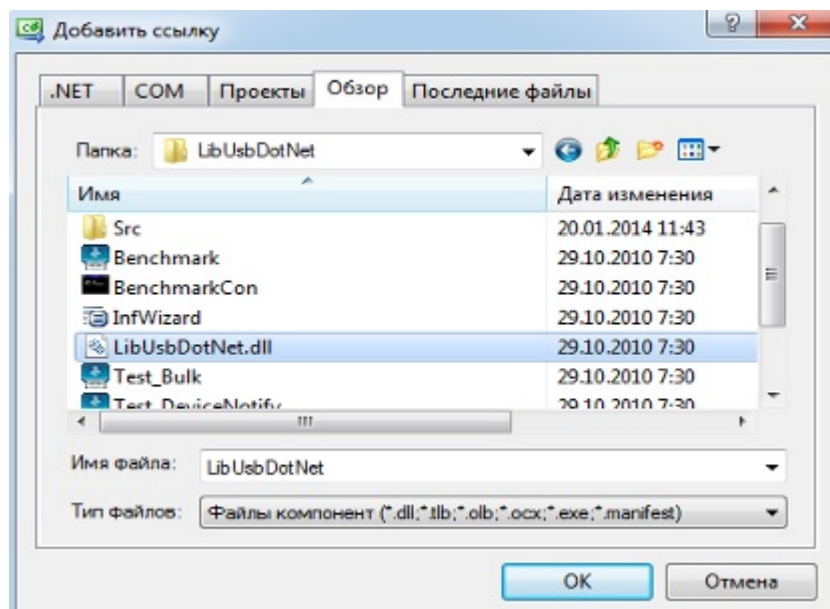
Однако главный цимус использования LibUsbDotNet вовсе не в том, что теперь легко и удоб

Запустите Microsoft Visual C# 2010 Express и создайте новый проект на основе Windows Form. Теперь нужно подключить к проекту библиотеку **LibUsbDotNet.dll**. В обозревателе решений нажмите правой кнопкой мыши на названии проекта, и выберите «Добавить ссылку».





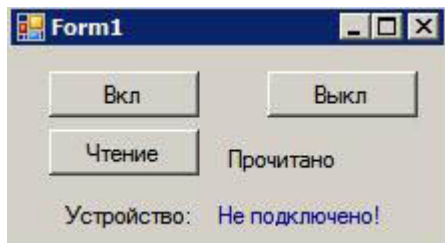
появится ещё одно окно



здесь нужно найти путь на диске, где находится библиотека LibUsbDotNet.dll (по умолчанию она устанавливается папку C:\Program Files\LibUsbDotNet, но лучше сделать копию файла DLL в рабочий каталог проекта. После подключения библиотеки её нужно объявить в проекте, для этого добавьте в главный модуль программы (файл Form1.cs) строки:

```
using LibUsbDotNet;  
using LibUsbDotNet.Info;  
using LibUsbDotNet.Main;
```

Перейдите к визуальному редактору формы, и приведите её приблизительно к такому виду (добавьте 3 кнопки Button и 3 текстовых метки Label):



Сделайте обработчик события загрузки формы. Он нужен для того, чтобы при старте программы происходила инициализация экземпляра класса LibUsbDotNet, через который осуществляется обмен с устройством USB. Перед началом обмена необходимо открыть доступ именно к нашему устройству, потому что к компьютеру может быть подключено несколько устройств USB HID, и необходимо уметь обращаться к каждому по отдельности. Для целей идентификации USB-устройств служат специальные идентификаторы, которые имеют абсолютно все устройства USB, это VID и PID.

Примечание. Иногда для идентификации устройства дополнительно используется уникальный с

VID это идентификатор производителя (Vendor ID), а PID — идентификатор устройства (Product ID). Наше USB-устройство имеет VID: 0x**16C0**, PID: 0x**05DF**, эти значения указаны в конфигурационном файле *usbconfig.h* (об этом файле мы уже упоминали) проекта микроконтроллера AS6. Чтобы ПО хоста обратилась к именно к нашему USB-устройству, нужно инициализировать объект MyUsbFinder такими же параметрами VID: 0x16c0, PID: 0x05df, как указаны в файле *usbconfig.h*. Для этого в область определения глобальных переменных класса Form1 добавьте следующий код:

```
public static UsbDevice MyUsbDevice;  
public static UsbDeviceFinder MyUsbFinder = new UsbDeviceFinder(0x16c0, 0x05df);
```



После того как мы определились с каким USB-устройством будем работать, можно к нему подключаться, и это удобно сделать в момент старта программы (открытия окна формы). Для этого выберите основную форму программы, и в редакторе свойств создайте обработчик события загрузки Form1\_Load. В теле обработчика введите следующий код:

```
private void Form1_Load(object sender, EventArgs e)
{
    MyUsbDevice = UsbDevice.OpenUsbDevice(MyUsbFinder);
    if (MyUsbDevice != null)
    {
        label2.Text = " подключено !";
    }
    else label2.Text = " не найдено !";
}
```

Сделайте обработчик события клика на кнопке button1 («Вкл»), для этого сделайте в визуальном редакторе на кнопке двойной щелчок, и добавьте в тело обработчика события код:

```
private void button1_Click(object sender, EventArgs e)
{
    // Передать пакет, который включает светодиод на макетной плате AVR-USB-MEGA16.
    UsbSetupPacket packet = new UsbSetupPacket((byte)(UsbCtrlFlags.RequestType_Vendor |
    UsbCtrlFlags.Recipient_Device | UsbCtrlFlags.Direction_Out), 1, (short)1, 0, 0);
    int countIn;
    byte[] data = new byte[1];
    MyUsbDevice.ControlTransfer(ref packet, data, 0, out countIn);
}
```

Для обработчика кнопки «Выкл» добавьте код:

```
private void button3_Click(object sender, EventArgs e)
{
    // Передать пакет, который погасит светодиод на макетной плате AVR-USB-MEGA16.
    UsbSetupPacket packet = new UsbSetupPacket((byte)(UsbCtrlFlags.RequestType_Vendor |
    UsbCtrlFlags.Recipient_Device | UsbCtrlFlags.Direction_Out), 1, (short)0, 0, 0);
```

```
int countIn;  
byte[] data = new byte[1];  
MyUsbDevice.ControlTransfer(ref packet, data, 0, out countIn);  
}
```

Код для обработки кнопки «Чтение»:

```
private void button2_Click(object sender, EventArgs e)  
{  
    //Получение данных от макетной платы AVR-USB-MEGA16 - состояние светодиода.  
    UsbSetupPacket packet = new UsbSetupPacket((byte)(UsbCtrlFlags.RequestType_Vendor |  
    UsbCtrlFlags.Recipient_Device | UsbCtrlFlags.Direction_In), 2, (short)0, (short)0,  
    int countIn;  
    byte[] data = new byte[1];  
    if (MyUsbDevice.ControlTransfer(ref packet, data, 1, out countIn) && (countIn == 1)  
    {  
        label3.Text = "Прочитано значение " + data[0].ToString();  
    }  
}
```

Обработчик события закрытия формы (завершение работы программы) гасит светодиод, если он горит:

```
private void Form1_FormClosed(object sender, FormClosedEventArgs e)  
{  
    UsbSetupPacket packet = new UsbSetupPacket((byte)(UsbCtrlFlags.RequestType_Vendor |  
    UsbCtrlFlags.Recipient_Device | UsbCtrlFlags.Direction_Out), 1, (short)0, 0, 0);  
    int countIn;  
    byte[] data = new byte[1];  
    MyUsbDevice.ControlTransfer(ref packet, data, 0, out countIn);  
}
```

## Как пакеты USB декодируются в firmware микроконтроллера

Прием и обработка данных на стороне микроконтроллера осуществляется в функции

*usbFunctionSetup* (находится в главном модуле VUSB.c проекта firmware AS6). Вот эта функция:

```
usbMsgLen_t usbFunctionSetup(uchar data[8])
{
    usbRequest_t *rq = (void *)data;

    if((rq->bmRequestType & USBRQ_TYPE_MASK) == USBRQ_TYPE_VENDOR){
        DBG1(0x50, &rq->bRequest, 1); /* отладочный вывод: печатаем наш запрос */
        if(rq->bRequest == CUSTOM_RQ_SET_STATUS){
            if(rq->wValue.bytes[0] & 1){ /* установить LED */
                LED_PORT_OUTPUT |= _BV(LED_BIT);
            }else{ /* очистить LED */
                LED_PORT_OUTPUT &= ~_BV(LED_BIT);
            }
        }else if(rq->bRequest == CUSTOM_RQ_GET_STATUS){
            static uchar dataBuffer[1]; /* буфер должен оставаться валидным привыха
            dataBuffer[0] = ((LED_PORT_OUTPUT & _BV(LED_BIT)) != 0);
            usbMsgPtr = dataBuffer; /* говорим драйверу, какие данные вернуть */
            return 1; /* говорим драйверу послать 1 байт */
        }
    }else{
        /* вызовы запросов USBRQ_HID_GET_REPORT и USBRQ_HID_SET_REPORT не реализованы,
        * поскольку мы их не вызываем. Операционная система также не будет обращаться
        * потому что наш дескриптор не определяет никакого значения.
        */
    }
    return 0; /* default для нереализованных запросов: не возвращаем назад данные хосту */
}
```

Наше устройство USB HID простейшее, и реагирует оно только на управляющие передачи (control transfer), которые проходят через конечную точку 0 (default control endpoint). По типу запроса (поле bRequest) декодируется направление передачи данных. Если CUSTOM\_RQ\_SET\_STATUS, то это данные, предназначенные для микроконтроллера. Данные декодируются и микроконтроллер выполняет заложенную там команду. В этом случае в самом первом по порядку принятом байте данных закодировано состояние светодиода — если там в младшем бите единица, то светодиод включается, а если нолик, то гаснет. Если же в поле bRequest принято значение CUSTOM\_RQ\_GET\_STATUS, то в ответ заполняется буфер текущим состоянием светодиода, и данные буфера отправляются обратно хосту. Все

очень просто, и при желании поведение кода можно легко переделать под свои нужды.

Видео, как это работает:

## Управление устройством USB HID на Windows

♦ +29

🔖 250



💬 20

Буду рад ответить в комментариях на вопросы и конструктивные замечания.

### Ссылки

1. V-USB.
2. Atmel Studio 6.
3. WinAVR.
4. LibUsbDotNet C# USB Library.
5. Visual Studio 2010 Express.
6. Исходный код для микроконтроллера и для ПО хоста.

*P. S. Толчком к познаниям стал сайт [microsin.net](http://microsin.net), где нашлось много информации по протоколам USB и практическому применению микроконтроллеров в USB-устройствах. Огромное спасибо! разработчику этого сайта за отзывчивость и помощь в вопросах касавшихся данной тематики.*

*В дальнейшем по возможности планирую сделать тоже самое, но на микроконтроллере с аппаратным интерфейсом USB.*

**Теги:** USB HID, V-USB, AVR, LibUsbDotNet, C#

**Хабы:** C#

### Редакторский дайджест

Присылаем лучшие статьи раз в месяц



Электронпочта



10

0

Карма   Рейтинг

**Антон Гончаров** @kubanbanzai

Пользователь

Реклама

## Комментарии 20



**Bonio**  
31 янв 2014 в 00:17

Для бóльшей стабильности микроконтроллер лучше запитать через LDO стабилизатор на 3.3v, резисторы R1, R2 увеличить до 150-220 Ом, R4 — не нужен.

Микроконтроллер лучше взять из серии 48/88/168/328, в отличии от Atmega8/16/32, при питании от 3.3v они гарантируют стабильную работу при 12MHz.

+1

[Ответить](#)

**SlavikMIPT**  
31 янв 2014 в 02:02

ну ведь не знаете же, а учите еще — нормальная схема для макета вполне — что по вашему стабильность контроллера? как он нестабильно работает? по даташиту ему 3.6V требуется, с учетом того, что частота довольно высокая 3.6-3.8 вполне нормально. Чем вам резисторы не угодили R1 R2? R4 нужен — по резисторам этим хост устройство определяет что к нему подключено — какой скорости устройство, болтающаяся в воздухе нога входа — отличный совет просто. А этот не гарантирует контроллер стабильной работы? Это все тут несущественно. Человек показал софтовую часть

0

[Ответить](#)

**Bonio**

31 янв 2014 в 02:44



что по вашему стабильность контроллера?

Не стабильность контроллера, а стабильность определения устройства компьютером. Говорю не безосновательно, специально экспериментировал.

по даташиту ему 3.6В требуется, с учетом того, что частота довольно высокая 3.6-3.8 вполне нормально.

Я не зря написал про серию 48/88/168/328, работа данных микроконтроллеров гарантируется на частоте 12MHz при напряжении всего 3.3v. Серия 8/16/32 на практике работает, но, по даташиту, при 12MHz требует больше, чем 3.3v.

Чем вам резисторы не угодили R1 R2?

Личные наблюдения. При увеличении номинала данных резисторов устройство успешно определяется на большем количестве компьютеров.

R4 нужен — по резисторам этим хост устройство определяет что к нему подключено

Хост определяет скорость (low-speed) по подтяжке резистором R3 линии D-. На линии D+ не будет неопределенных состояний, R4 тут без надобности.

Это все тут несущественно. Человек показал софтовую часть

Софтовая часть, в данном случае, бесполезна без аппаратной, и чтобы снизить вероятность траблов в процессе экспериментов, я поделился своими наблюдениями.

**+2**

Ответить

**KurilkaRymin**

31 янв 2014 в 04:41



Наверно R4 нужен для возможных статических разрядов на D+ при подключении устройства, не?

**0**

Ответить

**Derailed**

31 янв 2014 в 10:08



Вы правы, статья совсем не о схемотехнике, не о питающем напряжении и частоте кварца, и автора за это не в чем упрекнуть. Если бы к схеме предъявлялись какие-то специальные



требования в плане надежности и стабильности работы — тогда конечно, к выбору микроконтроллера, напряжению питания и рабочей частоте нужно уделять особое внимание. Кстати говоря, в макетной плате AVR-USB-MEGA16 микроконтроллер питается от 5V, и уровни сигналов USB создаются с использованием стабилитронов.



0

Ответить

**kubanbanzai**

31 янв 2014 в 14:41

Упор в статье сделан как правильно заметил «SlavikMIPT» и «Derailed» на софтовую часть, что касемо схемы то она разработана не мной, а (взята из пакета разработчика V-USB [1]) в качестве примера подключения контроллера по USB. Я сам новичок в этом деле как в C# так и в Atmel Studio 6 и микроконтроллерах. Для моей лично необходимости мне понадобилось соединить ПК с МК через USB, на тот момент был немного уже знаком с C# и Atmel Studio 6, но найти в интернете как это можно сделать с помощью C# и Atmel Studio 6 я так и не смог, не знаю почему, то ли я гуглить не умею то ли народ не хочет делиться своим опытом и знаниями. Я больше месяца потратил на то что бы это сделать ввиду моего малого опыта. И вот теперь всё получилось и теперь решил выложить, что бы такие как, я новички, не мучились долго.



+2

Ответить

**Alexeyslav**

31 янв 2014 в 15:54

Скорей дело было в отсутствии необходимости в соединении такой экзотики. Слишком уж разношерстный состав требуемого софта, и слишком нестандартные настройки нужны чтобы это все дело повторить. Почему к примеру WinAVR и 6-я студия? Неужели нельзя было обойтись исключительно AvrStudio с её штатным компилятором? Если уж на то пошло, то студия тут вовсе и не нужна, скомпилировать исходник можно было и без неё при наличии компилятора. Студия нужна только когда начнешь изменять исходник и отлаживать его, а делать это с таким проектом нереалистично — остается только одна функция студии — это интерпретация ошибок выдаваемых компилятором и установка курсора в место ошибки. Тяжеловат, однако, софт для такой функции.



0

Ответить

**kubanbanzai**

31 янв 2014 в 17:11

Просто изучения микроконтроллеров я начинал именно Atmel Studio 6, и поэтому как раз таки Atmel Studio 6 мне и нужна для развития моего дальнейшего проекта на основе того что уже сделано, поэтому я и замарочился именно с ней. Пробовал я со штатным компилятор AS6, при компиляции выдаёт ошибки, после их исправления проект компилируется, но после, заливаем прошивку в контроллер и он постоянно подключается и тут же отключается, моих познаний не хватает понять почему это так и это исправить мне так и не удалось. Если у кого это подучится просьба рассказать как если не сложно.



0

Ответить

**Alexeyslav**

31 янв 2014 в 15:46

Пожалуй, для поиграться только и пойдет. У меня большие сомнения насчет стабильной работы V-USB да и конкретного контроллера в этом качестве, ничего серьезней помогать диодом он уже не потянет. Да и разобраться новичку в коде V-USB никак не проще чем начать с честных контроллеров с аппаратной поддержкой USB.

V-USB когда-то создавался из-за дороговизны и редкости контроллера с аппаратным USB и как альтернатива для простых «игрушек», и ИМХО должен быть забыт — потому как нехорошая это тенденция тащить костыли в массы.

Ничего принципиально бы не изменилось, если для этой статьи взять контроллер с аппаратным USB — ведь смысл как я понял именно в драйвере и программе его использующей под x64 виндовс, это то ради чего и писалась статья.

И потом, один светодиод все умеют зажигать, отличным примером было бы произвольно зажигать 8 светодиодов или вовсе 64.



-2

Ответить

**kubanbanzai**

31 янв 2014 в 17:33

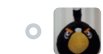


Сделан простейший пример как поморгать лампочкой для простейшего понимания новичкам, если вы профи то это не для Вас. И почему такая уверенность, что только для поиграться и поморгать одной лампочкой?.. Вот ссылка, на готовые проекты, посерьёзнее чем просто поиграться, с помощью библиотеки V-USB [www.obdev.at/products/vusb/prjobdev.html](http://www.obdev.at/products/vusb/prjobdev.html). А насчёт контроллеров с аппаратной поддержкой не спору что они лучше, но однако также народ не сильно делиться готовыми примера проектов. На [youtube](http://youtube) полно видео как люди мастерят USB устройства на контроллерах со встроенным интерфейсом, но там лишь демонстрация своего проекта типа какой я «умный». А выложить исходники и объяснить как и что ты сделал по шагам никто не хочет все жмутся и мне как новичку в этом деле это очень сложно. Если у вас есть готовый проект на контроллере с USB интерфейсом и с описанием как и что делать по шагам, пожалуйста поделитесь! Буду очень признателен я и многие другие начинающие.



+1

Ответить

**Alexeyslav**

31 янв 2014 в 19:48



Так ведь ютуб не для выкладывания исходников, поэтому естественно их там нет.

Учится по исходникам чужих проектов сомнительного качества — удовольствие ниже среднего, кроме того это способствует перетаскиванию ошибок из проекта в проект.

Ваш пример компиляции V-USB тому пример — если она построена на особенностях конкретного компилятора, то так и будет. По-хорошему, нужно было выяснить причину возникновения ошибок а не исправлять их не понимая ничего а для этого нужно перечитать всю

библиотеку вручную и сверится с документацией по протоколу USB что все задано так как нужно. С точки зрения компилятора программа теперь не содержит ошибок, но с точки зрения программы вероятно теперь написана полная чушь(помните высказывание, что не всякая программа что скомпилировалась является рабочей?). Возможно, какие-то константы теперь приравнены нулю, какие-то макросы срабатывают не так как нужно и т.д. Порой проще переписать с нуля, чем исправлять такой трупик.

Это вот хорошая идея, взять книжку по USB в другую руку — даташит на контроллер и начать решать проблему, но у меня постоянно не хватает времени.  
Вот кстати по 4-й ссылке на гугле нашел библиотеку под USB для AVR.



-1

Ответить

**kubanbanzai**

31 янв 2014 в 20:11



я не говорю про переписывание чужих проектов, просто когда делаешь проект сам, а ещё и если ты только осваиваешь эти дебри, то иногда попадаешь в тупик и топчешься, топчешься, на месте и спросить то особо не у кого, а когда есть готовый пример то можно хотя в нём подсмотреть как реализован тот или иной протокол вот в чём смысл. А по про библиотеку LUFA я уже читал и кстати следующим моим шагом будет её изучение.



0

Ответить

**Derailed**

1 фев 2014 в 16:19



«Пожалуй...» «у меня большие сомнения...» Уважаемый @Alexeyslav, если Вы не разбираетесь в том, о чем пишете, то это еще не повод троллить беднягу автора статьи. Мне за Вас стыдно. Лучше займитесь чем-нибудь полезным.



+3

Ответить

**serafims**

10 дек 2014 в 14:55

Скажите, а вам не попадалось решение, позволяющее микроконтроллеру быть одновременно клавиатурой стандартной и вот таким вот кастомным HID устройством, которое принимает данные, скажем, подсвечивая активные кнопки, а при нажатии генерируется стандартная посылка hid клавиатуры..?



0

Ответить

**Alexeyslav**

10 дек 2014 в 16:17



В рамках одного канала это разве что расширить набор команд которые принимает клавиатура или мастерить программный хаб и подключать к нему два виртуальных девайса.



+1

Ответить

**serafims**

10 дек 2014 в 17:19



спасибо за ответ!



0

Ответить

**imwode**

11 дек 2014 в 10:52



Просто в USB-дескрипторе описываются два устройства, примеров да же в магазине полно, например клавиатуры с трэкболами.

00h Device Unspecified[34] Device class is unspecified, interface descriptors are used to determine needed drivers



0

Ответить

**Massaraksh147**

31 янв в 09:54

Много времени прошло, но всё же попробую задать вопрос: А без драйвера и на Delphi можно получить доступ к самодельному HID-устройству? Интересует посылка нескольких байт на него и получение ответа.



0

Ответить

**Alexeyslav**

31 янв в 22:21



Без драйвера ничего не работает в принципе. Но, можно использовать один из классов HID для которых драйвер уже изначально есть в виндовс. Так бесперебойники подключаются.



0

Ответить

**Massaraksh147**

1 фев в 01:36



Понятно, я и имел в виду встроенный драйвер. Но у меня вопрос скорее практический, нежели теоретический. В теории я знаю - как, на практике на Delphi не получается.



0

Ответить



Только полноправные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.

## Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ

**0serg**

6 часов назад

### Ортографическая проекция в фотографии и прочие интересные фокусы с объективами

**Средний**

29 мин



1.8K

Обзор

**+47**

36



5

**divolko3**

23 часа назад

### Один из старейших языков программирования COBOL, похоже, вскоре уйдёт с рынка. И поможет ему в этом IBM



3 мин



7.3K

**+22**

9



35

**ru\_vds**

4 часа назад

### Лучший софт для автоматизации скриншотов

**Простой**

6 мин



1.7K

Кейс

**+16**

9



8

**vladkorotnev**

5 часов назад

### Что внутри у однорукого бандита? Разборка, моддинг, и написание кейгена. [Part 1]

**Простой**

8 мин



1.3K

Обзор

 +10 6 2**headmusic**

3 часа назад

## Электростаты — электростатические наушники, как это работает?



Средний



6 мин



1K

Из песочницы

 +9 7 0**Oksana\_Nedvigina**

23 часа назад

## Чтение на выходные: «Психбольница в руках пациентов. Алан Купер об интерфейсах»



Простой



4 мин



1.9K

Обзор

 +9 23 12**Underskyer1**

2 часа назад

## Теория типов



Средний



43 мин



698

Из песочницы

 +8 10 0**AlexChroot**

22 часа назад

## Переписываем наш Object Cloud



Средний



8 мин



658

Тutorial

 +6 2 0



**AndreyKotlov**

17 часов назад

## Что такое арбитраж? Передовые технологии торговли на примере криптобиржи

**Сложный**

31 мин



1.9K

Аналитика

**+5**

39



7

**mssqlhelp**

21 час назад

## Новое в SQL Server 2022: Контейнерные группы доступности

**Сложный**

5 мин



1.2K

Обзор

**+4**

8



1

## Spring, микросервисы, ванлав — участвуем в новом сезоне Java

Турбо

Показать еще

МИНУТОЧКУ ВНИМАНИЯ



Проверь свою SQL-экипировку в тесте для аналитиков



Боги приковали Промокодуса за слишком щедрые скидки

## КУРСЫ



Аналитик BI

5 сентября 2023 · 87 000 ₽ · Нетология



Иллюстрация и интерактивная графика

5 сентября 2023 · 127 440 ₽ · Нетология



Системный анализ. Разработка требований к ПО - в группе

5 сентября 2023 · 36 000 ₽ · STENET school



Аналитик данных

5 сентября 2023 · 100 020 ₽ · Нетология



Веб-дизайнер: расширенный курс

6 сентября 2023 · 126 180 ₽ · Нетология

Больше курсов на Хабр Карьере

Реклама

<https://habr.com/ru/articles/210736/>



[Настройка языка](#)

[Техническая поддержка](#)

[Вернуться на старую версию](#)

© 2006–2023, Habr

















## ЧИТАЮТ СЕЙЧАС

---

Единственное в мире число-субфакторион — это 148349. Что это такое?

 121K  75

---

«Кандидат имеет право задавать уточняющие вопросы», или Доводим интервьюера до нервного срыва

 120K  222

---

YandexGPT тоже провалил тест на ручник

 26K  61

---

Ква! Как писали код во времена Quake

 13K  13

---

Канадские операторы разорвут соглашение о роуминге с российскими

 5.2K  85

---

Spring, микросервисы, ванлав — участвуем в новом сезоне Java

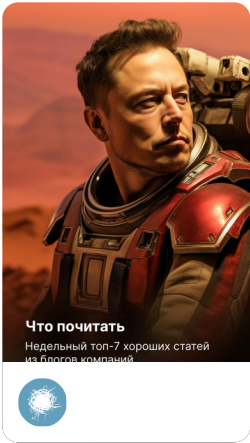


Турбо

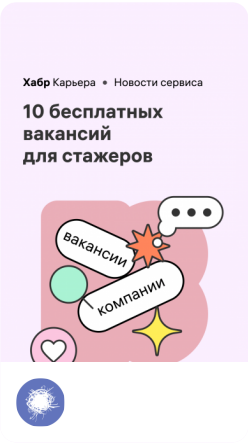
ИСТОРИИ



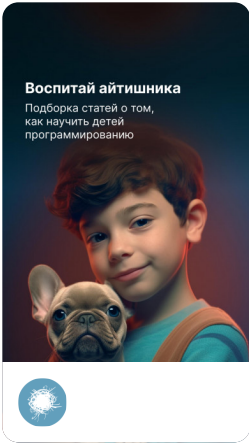
Перевернуть календарь и добавить событие



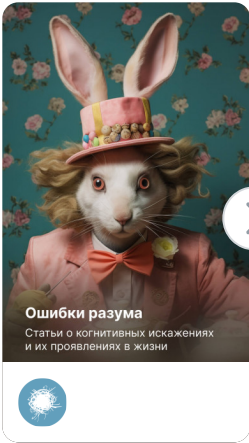
Топ-7 хороших статей из блогов компаний



Дарим вакансии для стажеров



Как учить детей программированию



Когнитивные искажения

РАБОТА

Программист C# удаленно  
105 вакансий

Все вакансии

Реклама