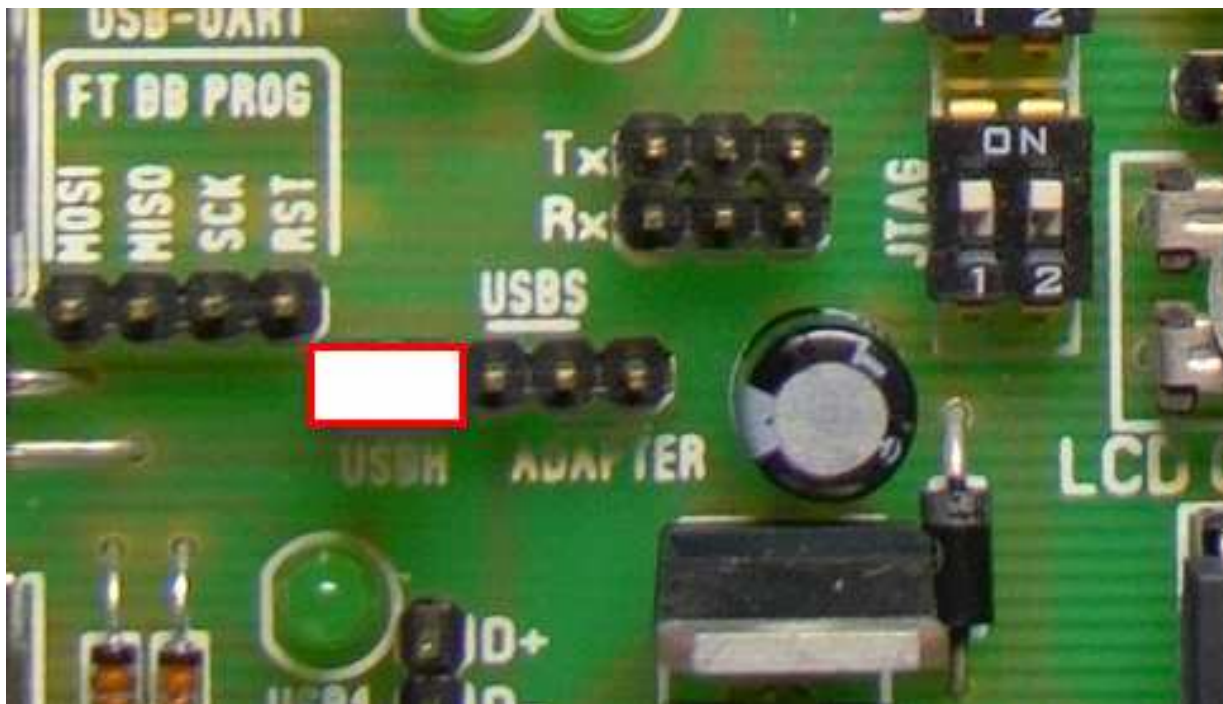


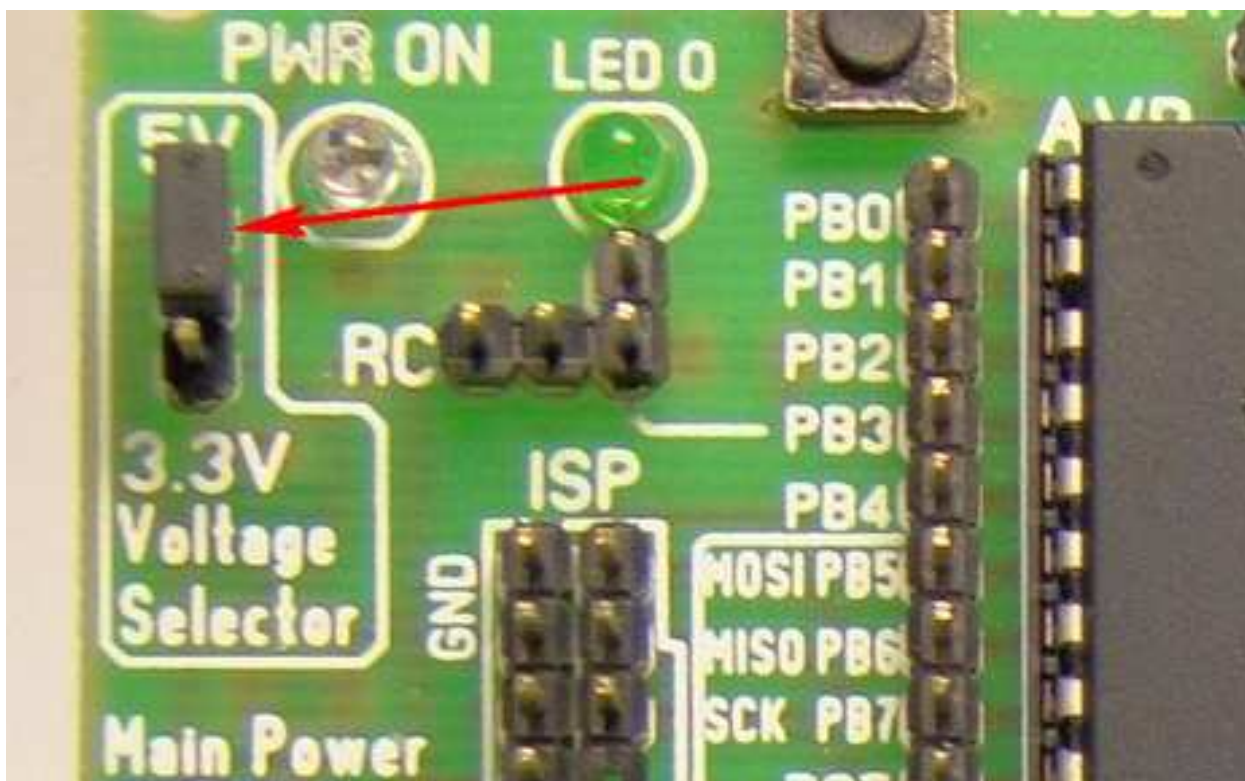
# PinBoard v1.1. Быстрый старт.

## Подача питания

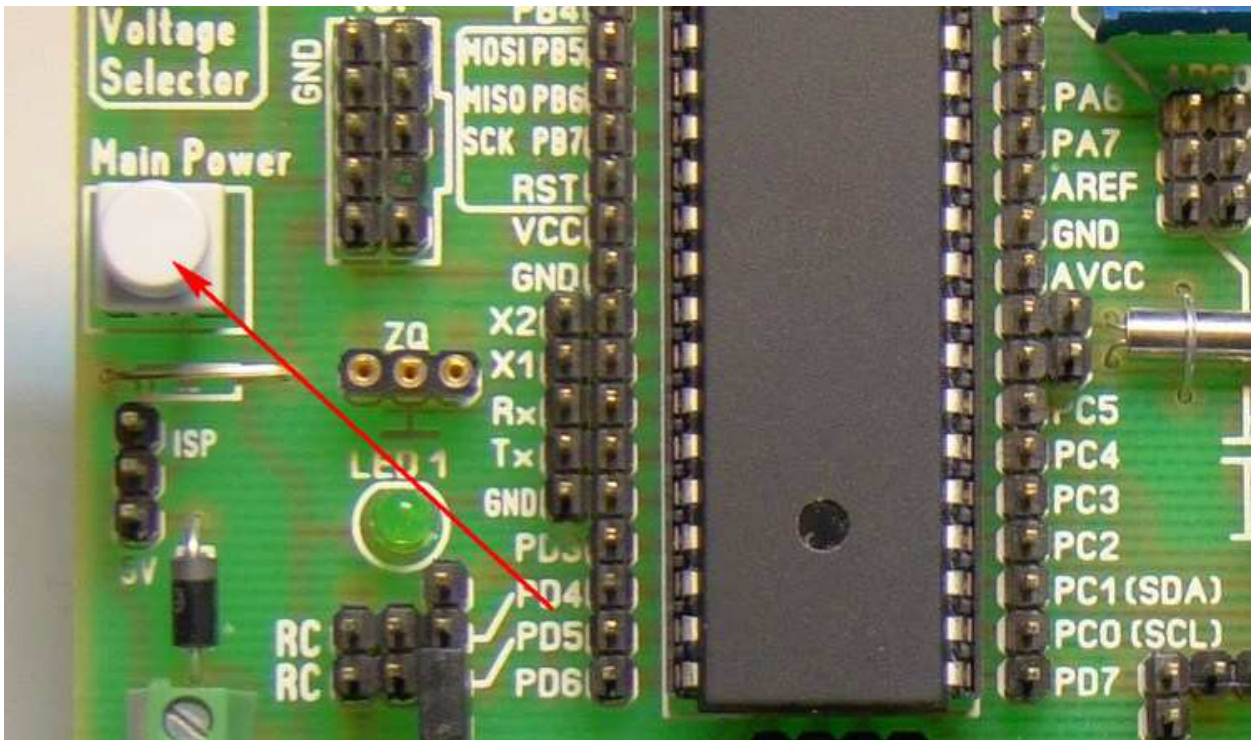
Подключите питание на Pinboard. В начале будем питать плату от USB шины, так что воткните джампер выбора питания в положение USBH



Теперь надо выбрать питающее напряжение. Для этого установите перемычку на селекторе питания в положение 5V:



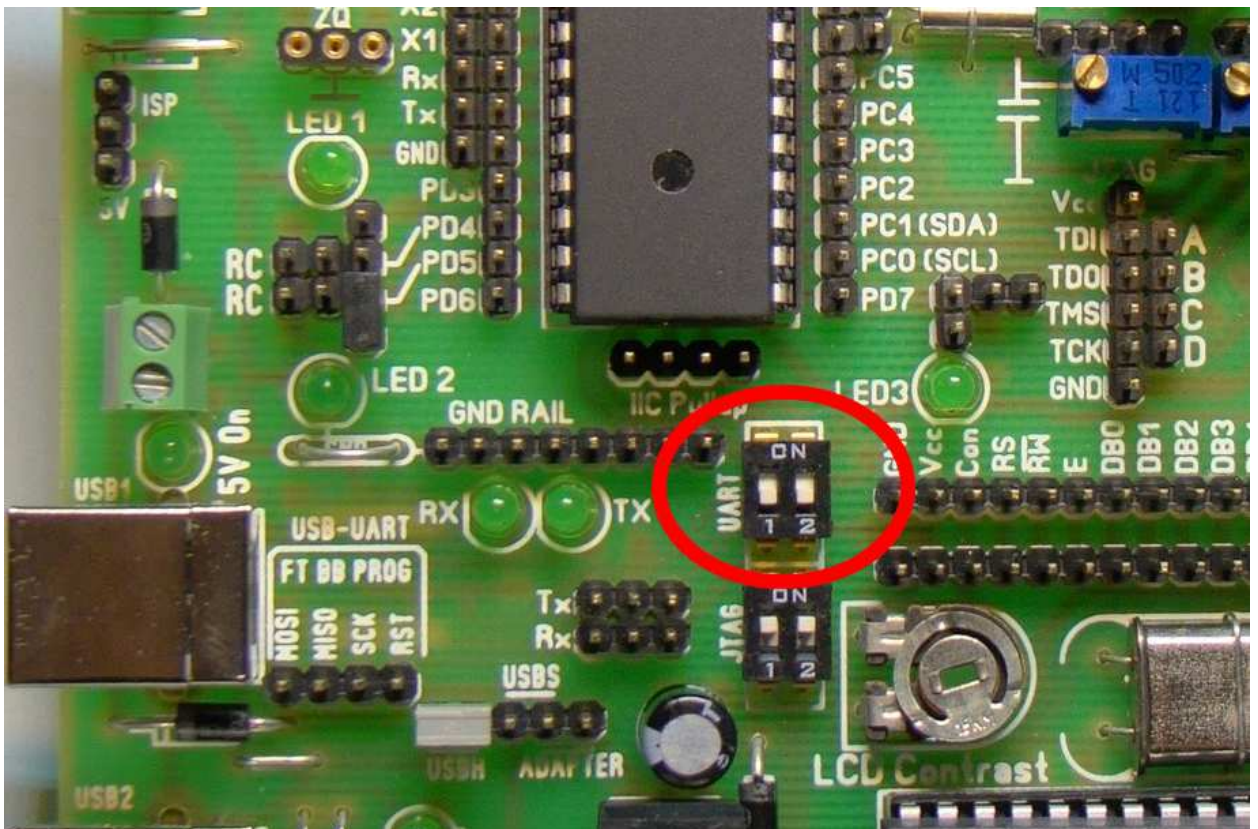
Кнопкой Main Power подайте питание на плату.



При этом должен загореться красный светодиод PowerON.

### Подключение демплаты к компьютеру

Первым делом нужно проверить интерфейс. Он свяжет нашу демоплату с компьютером. Для этого поставь переключатель UART и JTAG (оба рычажка у каждого) в положение OFF тем самым мы **отключим** его от контроллера. Надо следить за тем, чтобы эти переключатели не были **одновременно в положении ON**. Ничего страшного не будет, но работать в этом случае интерфейс не будет – из-за конфликта источников. Либо UART, либо JTAG, либо оба OFF, иного не дано.

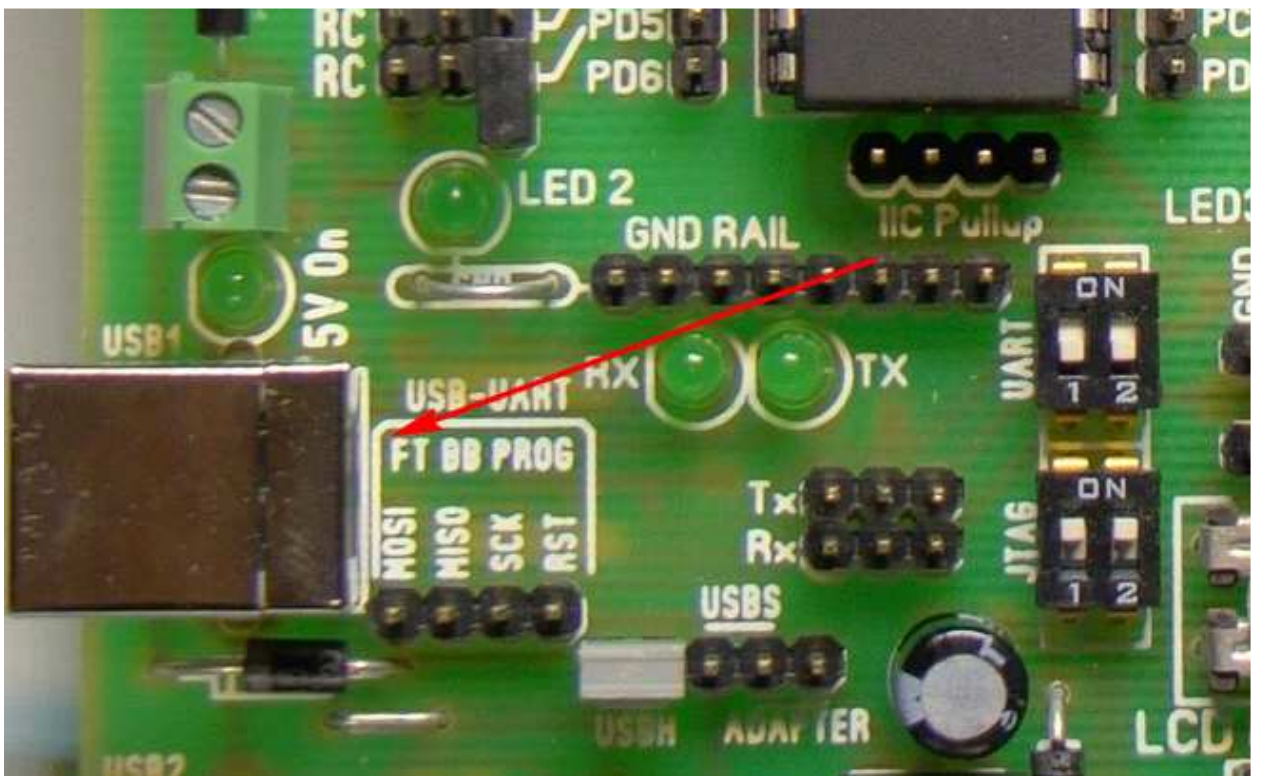


Также потребуется кабель USB-AB его можно купить в любом магазине торгующем компьютерами и оргтехникой. Красная цена ему 100рублей. Кабель вот такого вида:



С другой стороны у него стандартный USB штырь как у флешки обычной. У принтера обычно такой же кабель. Можно временно использовать его.

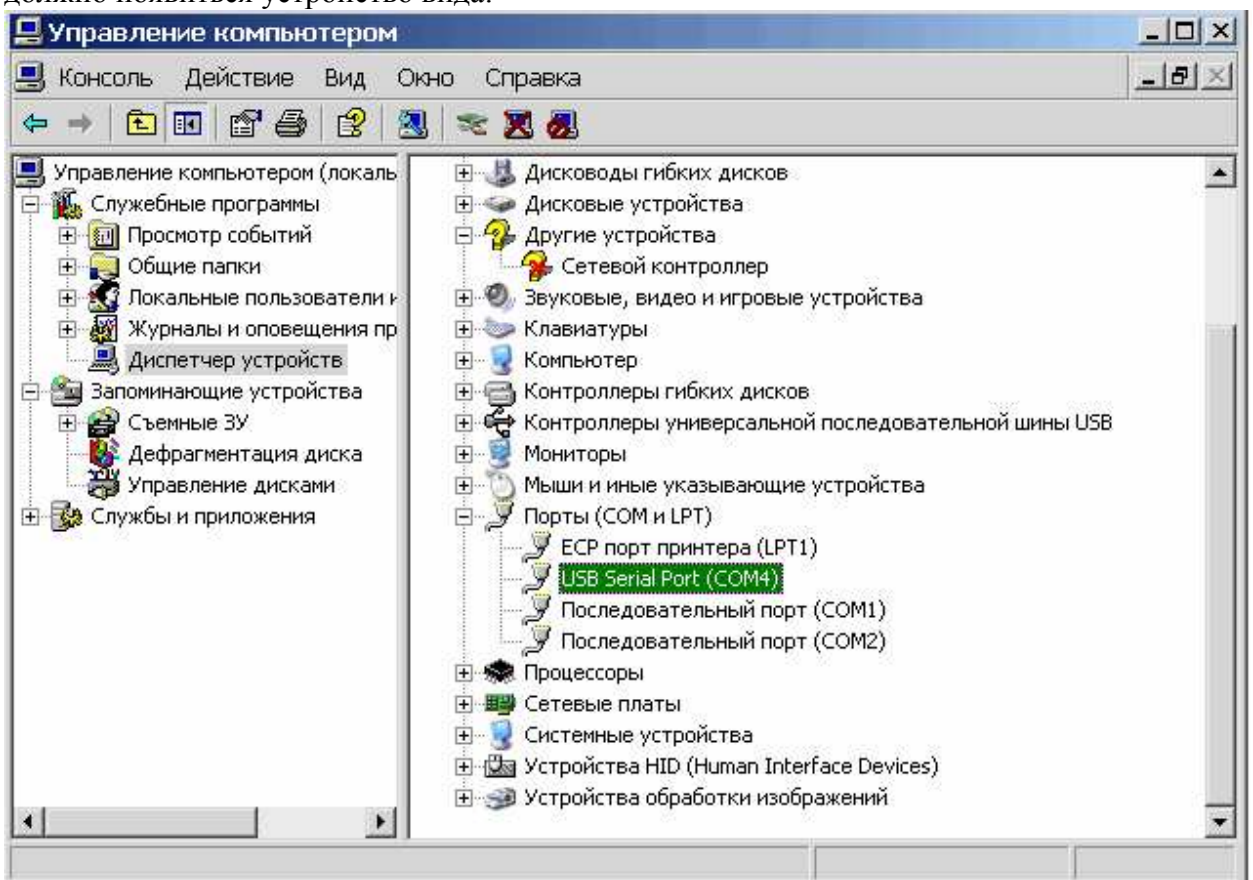
Подключаем кабель в компьютер и вставляем его в верхний (аппаратный) USB разъем платы PinBoard:





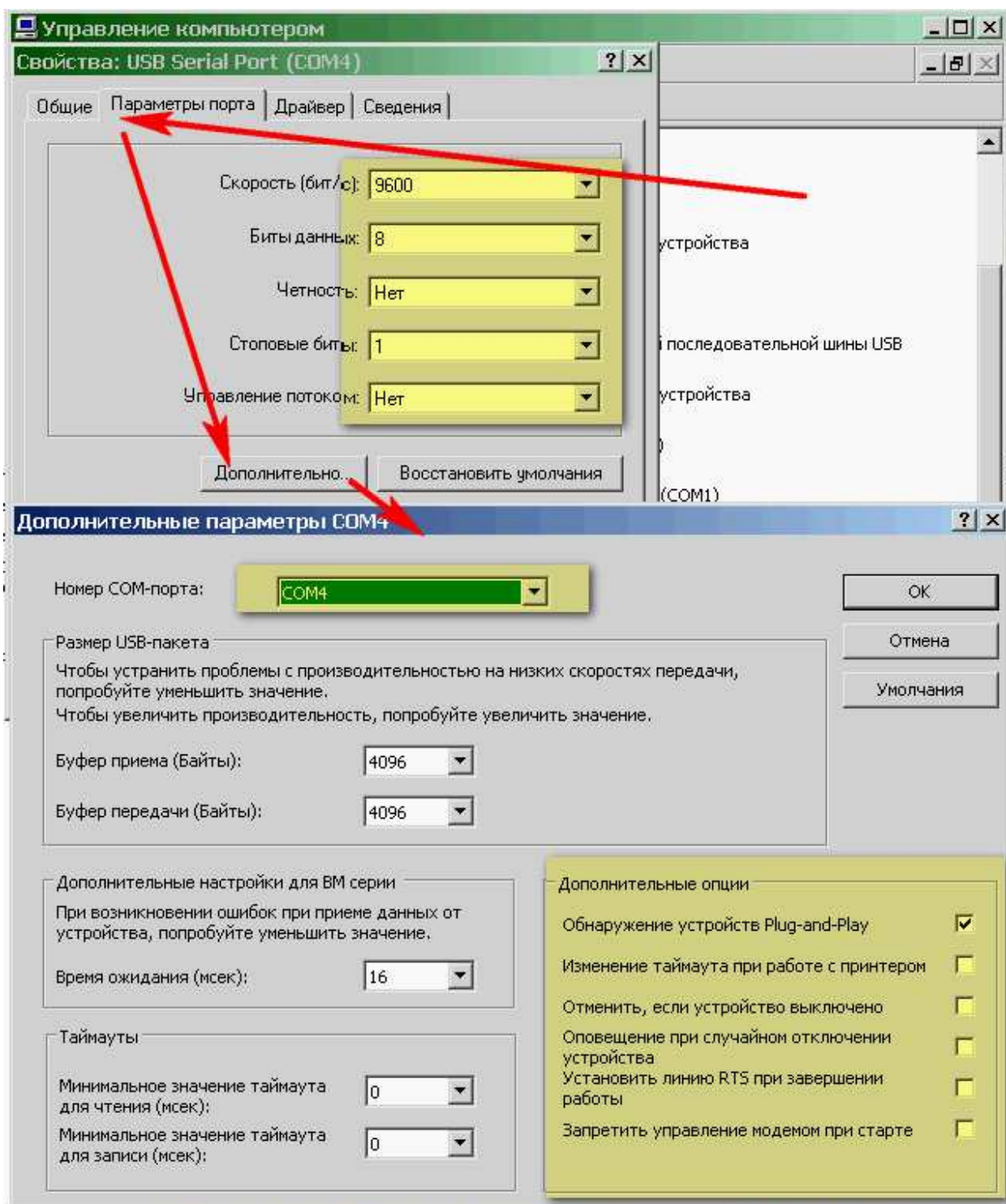
При этом сразу же компьютер должен отrapортовать, что найдено новое устройство и попытаться установить на нее драйвера. Обычно все сразу определяется нормально, но может потребоваться драйвер. Тогда вытащите кабель и сначала установите драйвера FDTI. Скачать их можно с сайта [ftdichip.com](http://ftdichip.com)

После установки драйвера снова воткните шнур и плату и на этот раз у вас в системе должно появиться устройство вида:

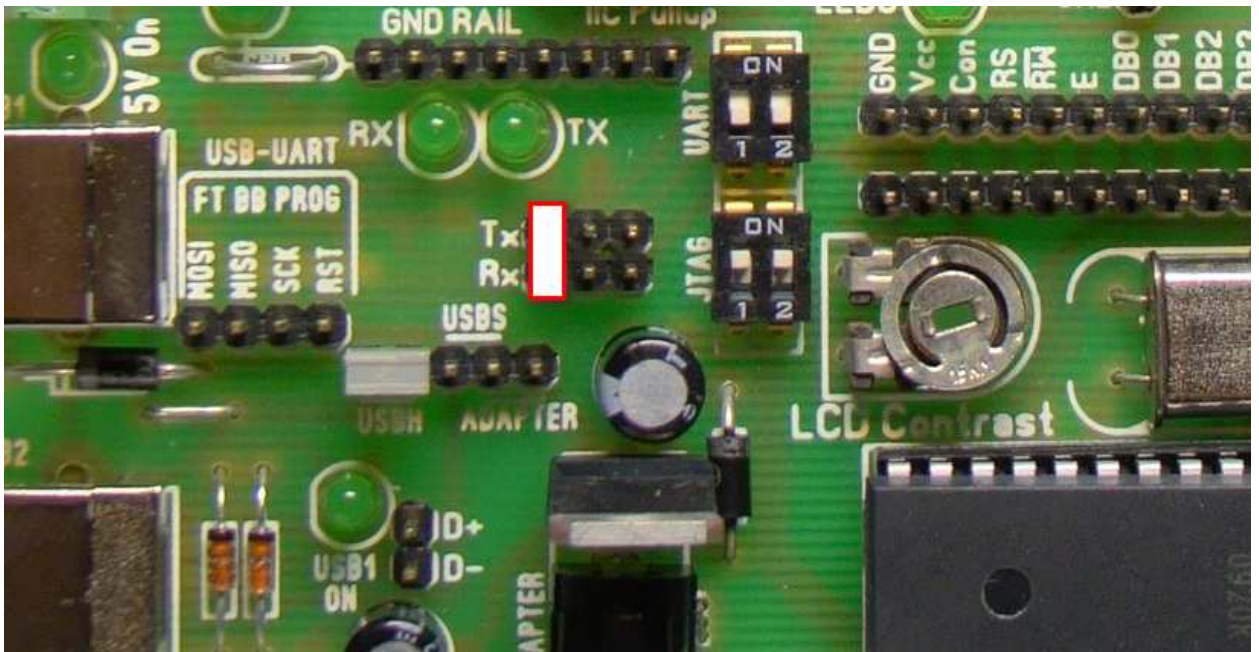


Зайдите в настройки и проверьте номер виртуального COM порта. Он должен быть в диапазоне от 1 - 4. Остальные настройки по дефолту.

Это важно, т.к. AVRProg не обнаружит загрузчик если у порта будет номер выше четвертого, т.к. он опрашивает только первые четыре порта.

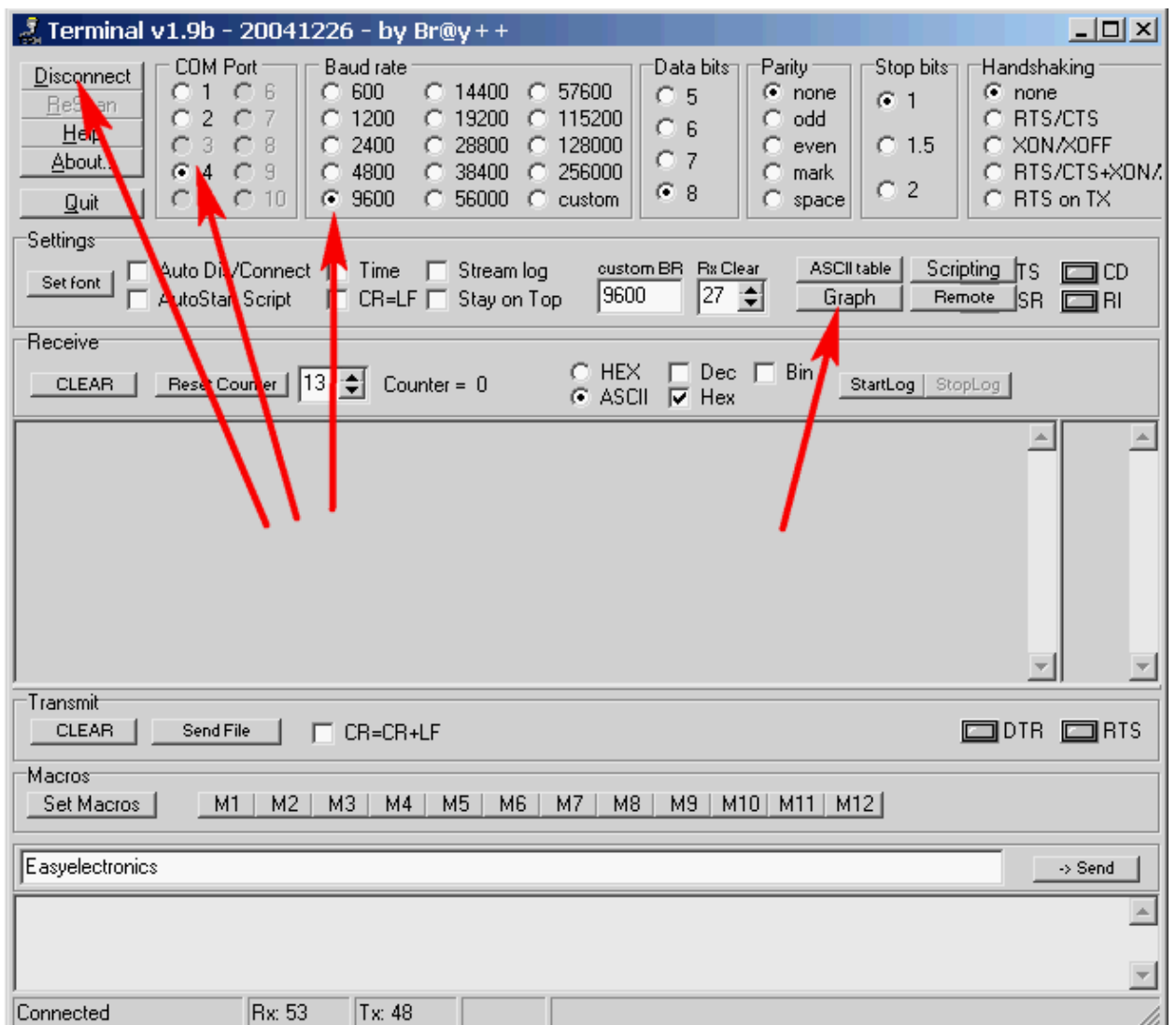


Теперь надо проверить работу интерфейса. Возьмите один джампер и наденьте его на выводы RxD и TxD, что возле USB разъема (смотри фото):

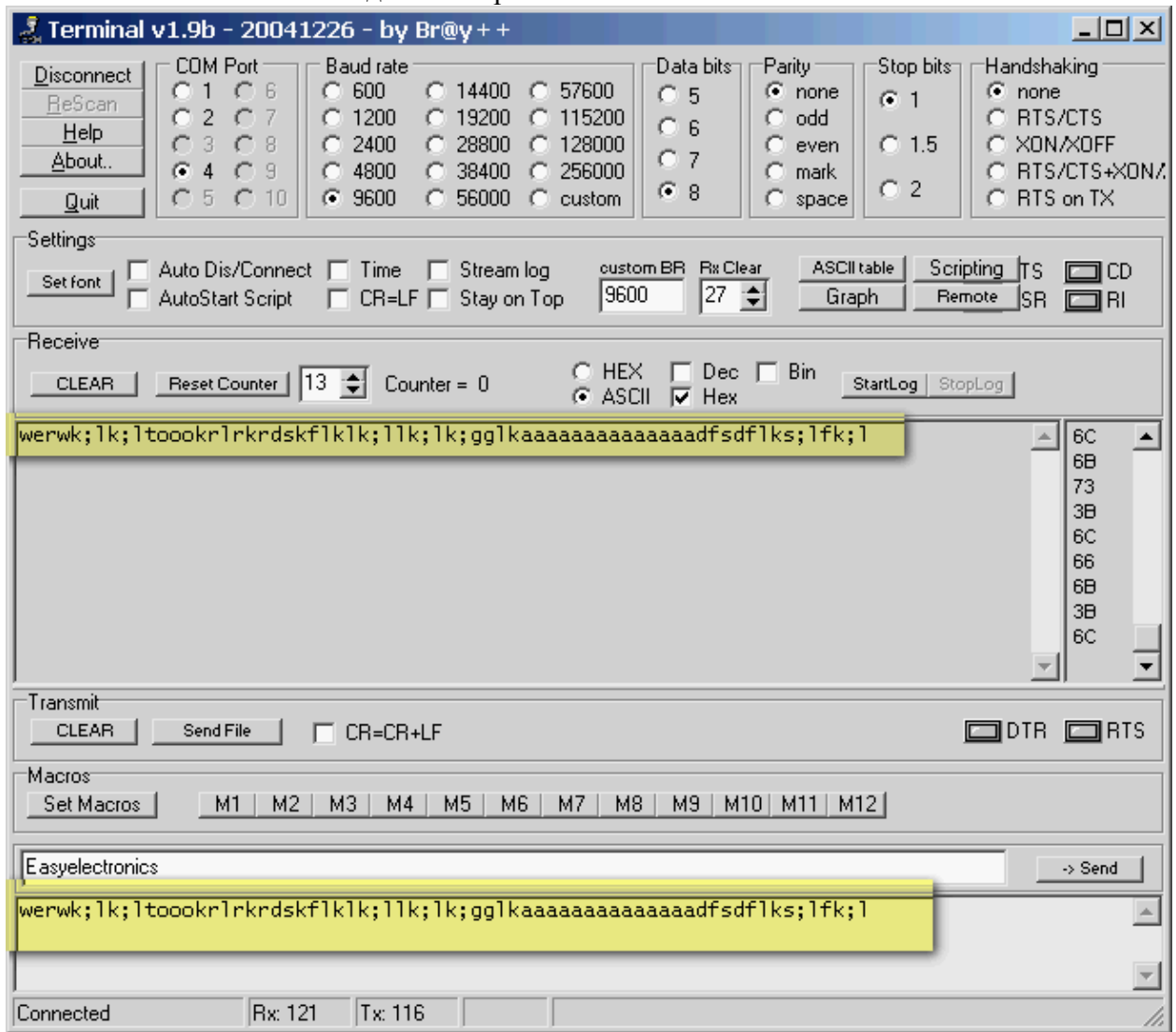


Переключатели JTAG и UART поставьте (если еще не сделали это) в положение OFF

Запустите терминальную программу, например Terminal v1.9b и подключитесь к нашему виртуальному COM порту.



В нижнем окне мы отправляем данные, в верхнем окне у нас принятые байты. Так как на плате TxD и RxD замкнуты джампером то уходящие байты должны возвращаться в точности. Что можно наблюдать на скриншоте:



Если этого не произошло, то возможно у вас неправильно установились драйвера на FTDI, либо вы неправильно установили переключку. В любом случае, при правильной работе FTDI при передаче данных должны моргать светодиоды RX--TX возле коннектора FTDI.



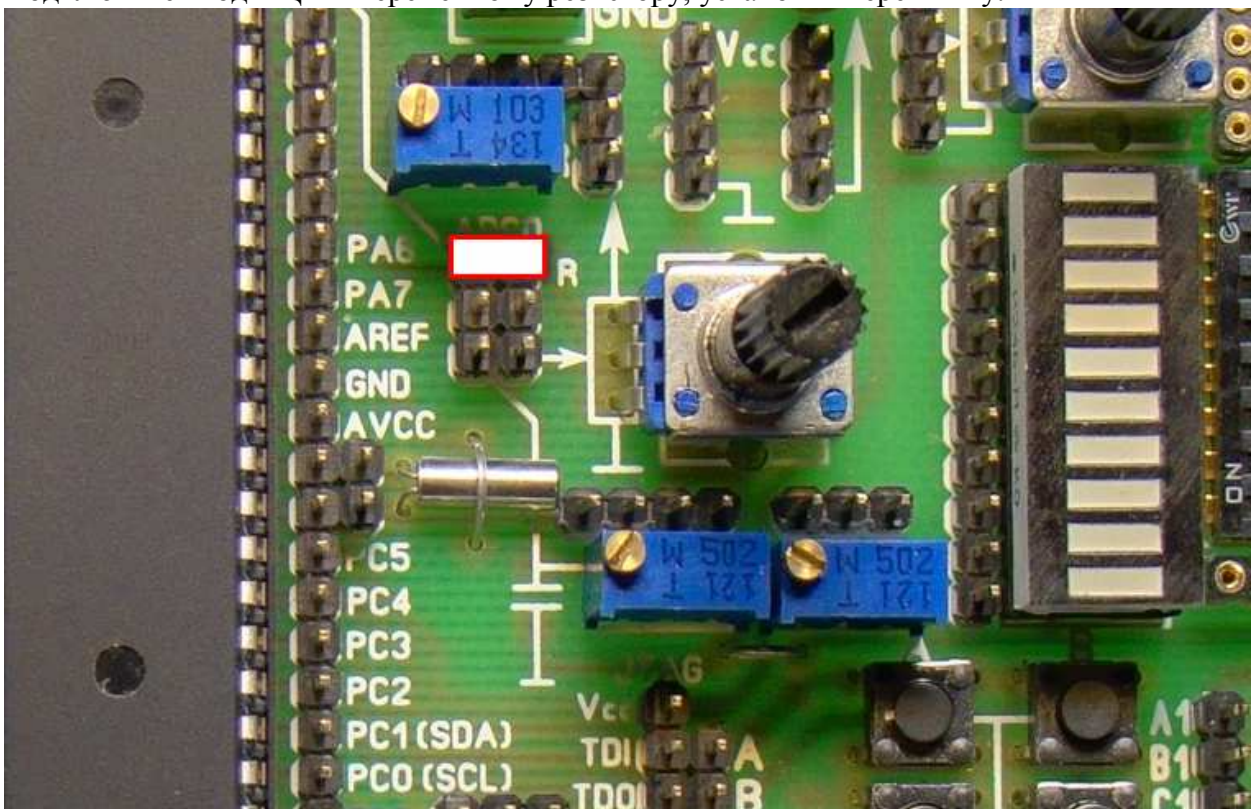
Но светодиоды не будут моргать если драйвер установлен не правильно или не открыт COM порт в компьютере.

#### **Подключение контроллера.**

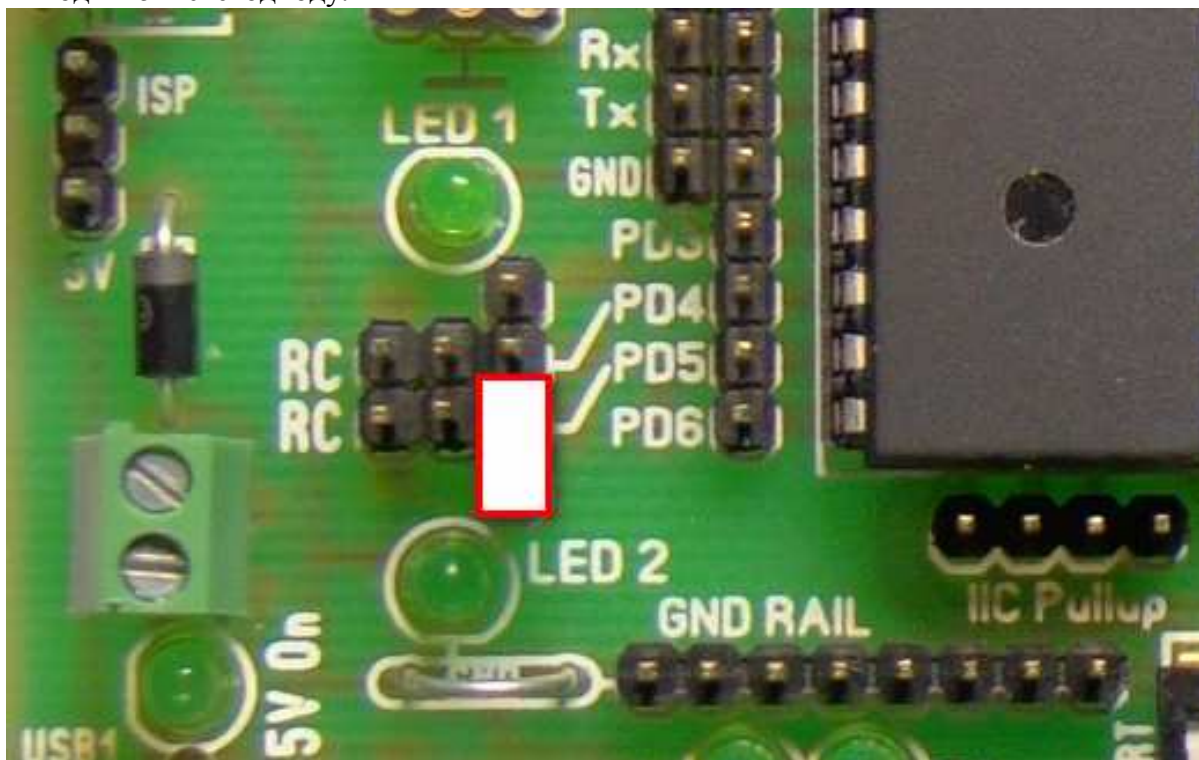
Затем переключатель **UART** (который описан в самом начале) поставим в положение ON (JTAG пусть так и остается в OFF)

С самого начала микроконтроллер прошит демопрограммой которая шлет в компьютер показания с канала 0 АЦП

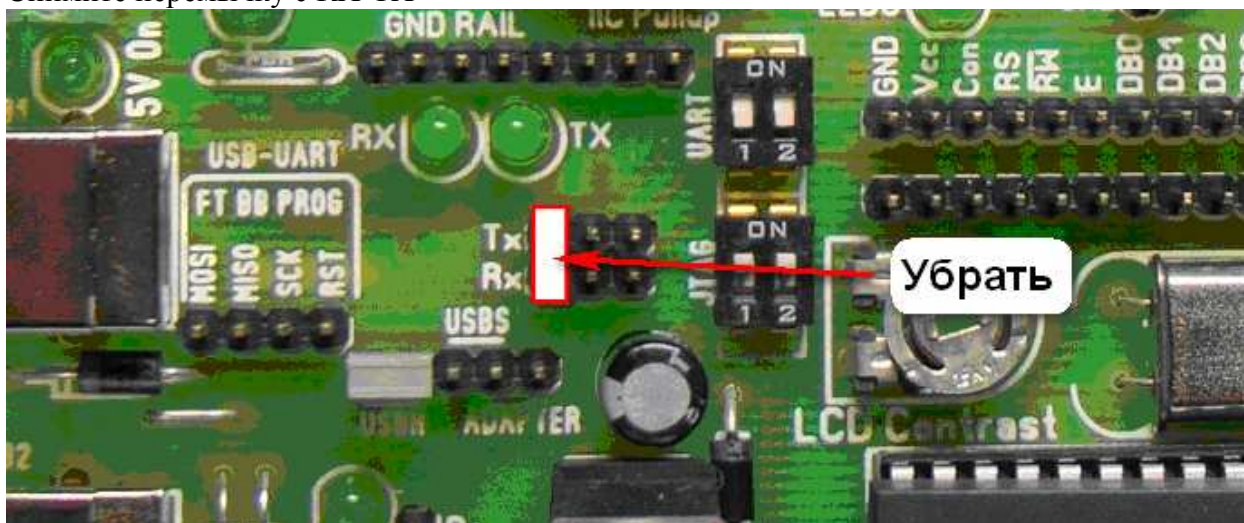
Подключите вход АЦП к переменному резистору, установив переключку:



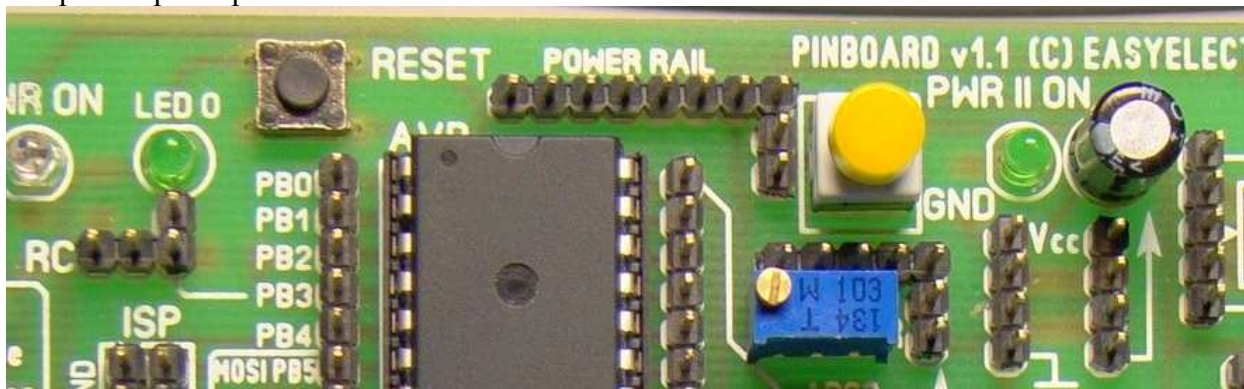
Также установите перемычку на индикаторный светодиод LED2, подключив тем самым вывод PD5 к светодиоду.



Снимите перемычку с RX-TX

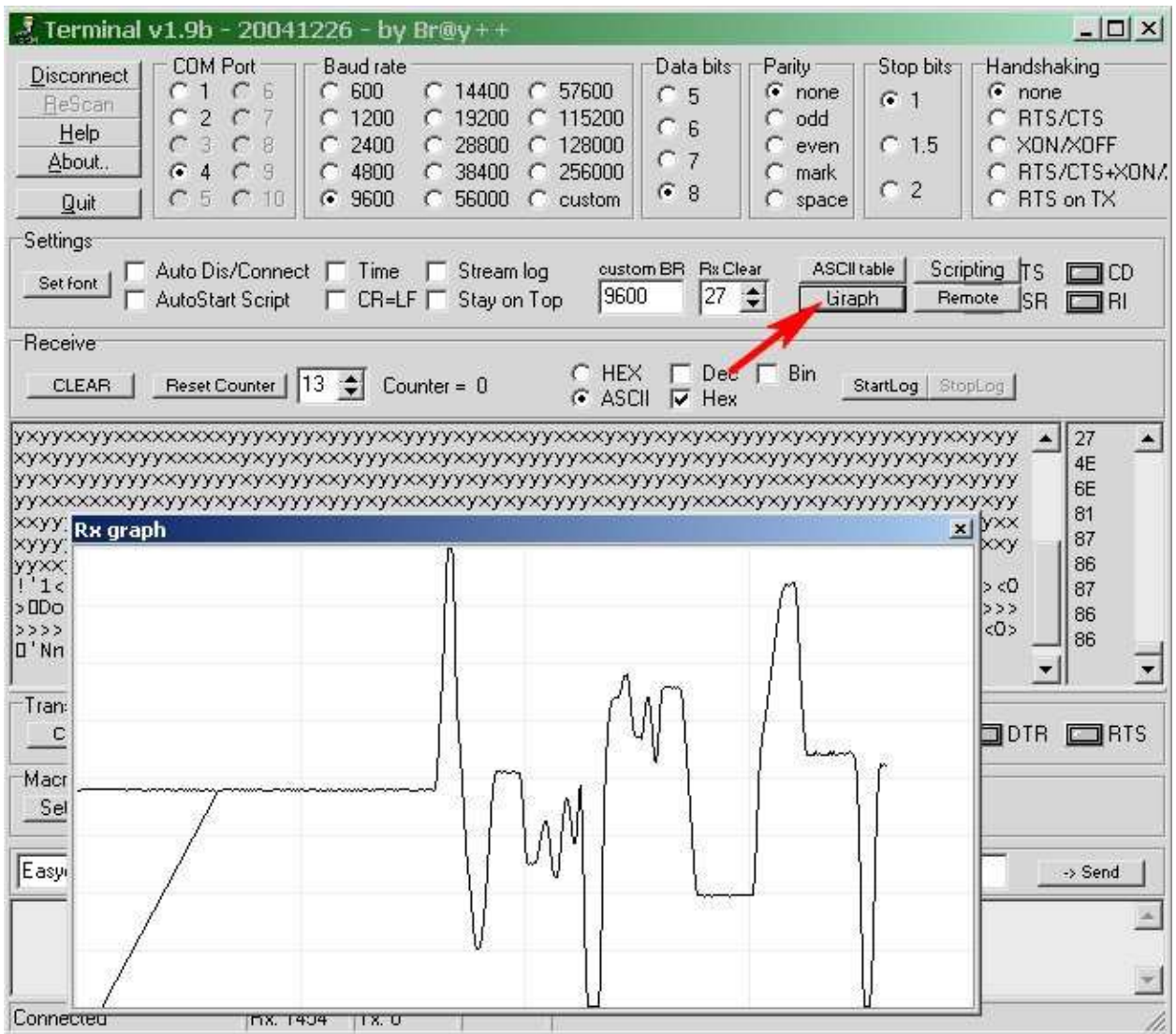


После нажмите кнопку RESET для сброса микроконтроллера. Она находится над главным микроконтроллером.



С этого момента контроллер начинает выполнять загрузку в него демопрограммы:



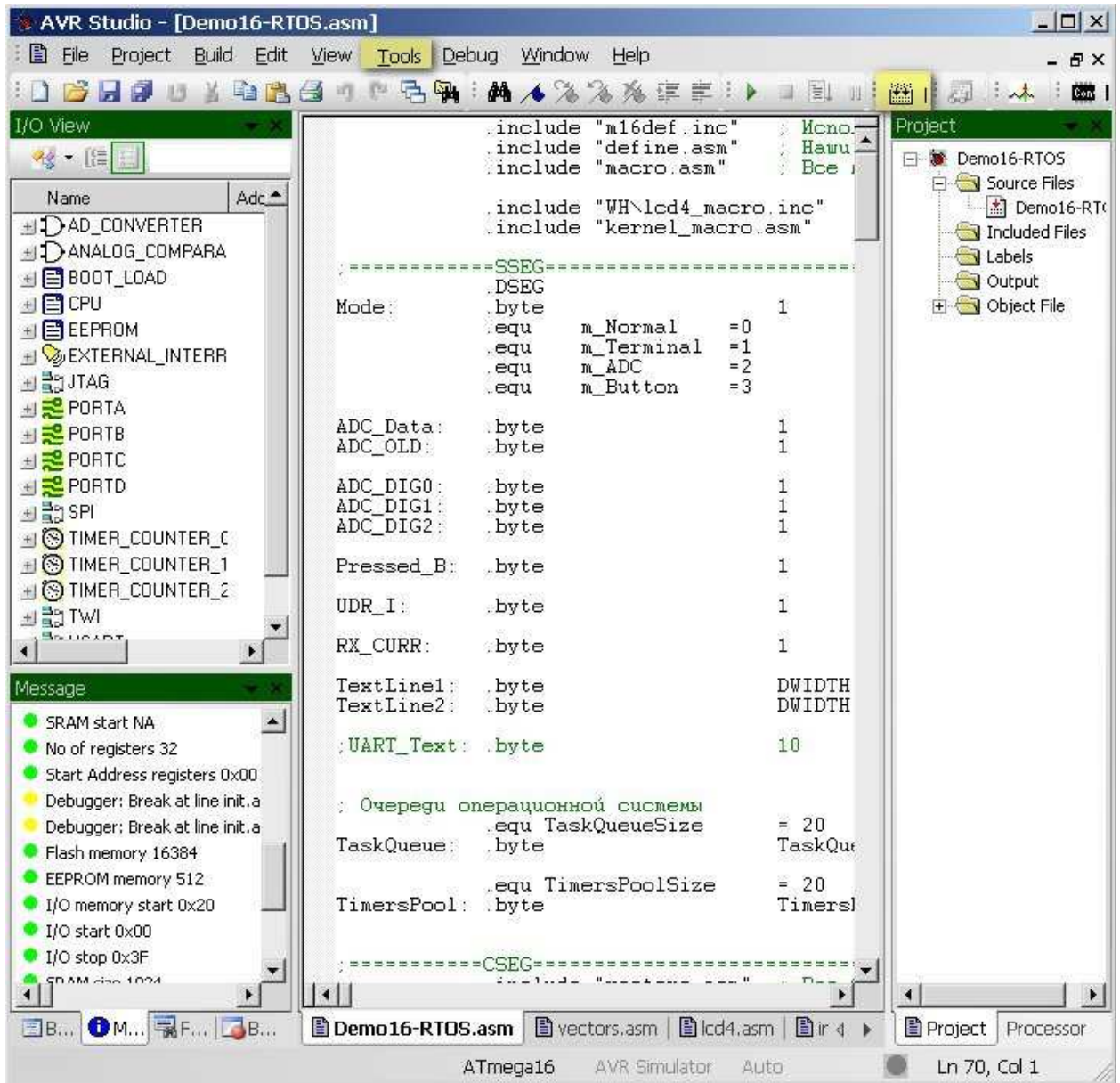


Вращая в разные стороны вал потенциометра (который подключен к входу АЦП) мы меняем напряжение на входе АЦП, о чем нам тут же сигнализирует график в терминальной программе

Отлично. Все проверено, все работает как надо. Теперь покажу как менять программу посредством бутлоадера.

### Загрузка собственной программы через bootloader:

Запускайте **AVR Studio** (скачать ее нужно с сайта [atmel.com](http://atmel.com) заполнив регистрационную форму. Регистрация бесплатная, только для статистики. Пишите туда что хотите) и открывай файл со второй демопрограммой. Той что в части 3 документации на Pinboard v1.0 (разница между демоплатами тут уже не принципиальная). В окнах студии откроется код программы. Можно его скомпилировать нажав на F7.



Закройте терминальную программу, предварительно нажав Disconnect, чтобы освободить доступ к порту.

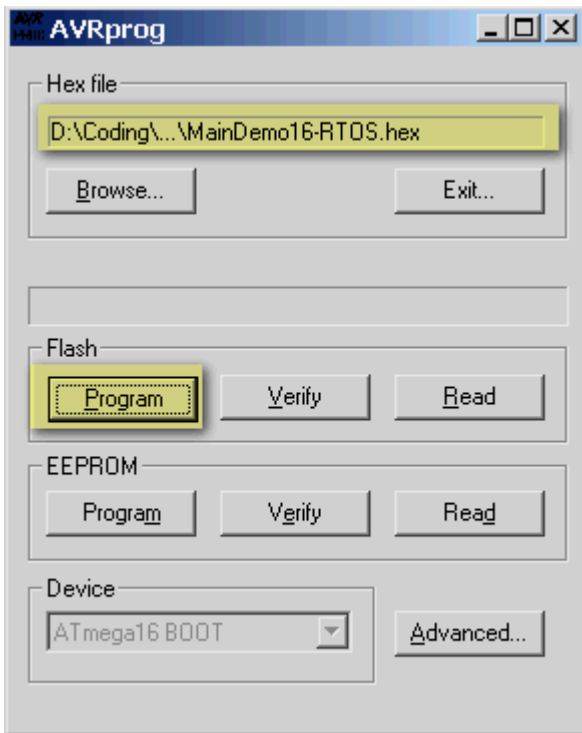
**Это важно! Иначе AVRProg не увидит бутлоадер.**

Теперь найдите в AVR Studio программу AVRProg, она находится в меню Tools -- avrprog, но не запускайте ее.

Дальше действуйте быстро:

- Нажмите RESET на плате (квадратная такая кнопка возле микроконтроллера)
- Пока горит зеленый светодиод LED2 (работа бутлоадера) нужно успеть вызвать AVRProg.

Должно получиться вот так:



Все, можете выбрать в первой строке hex файл для прошивки и залить его в память контроллера, нажав на кнопку «Program» в секции Flash. А загрузив Hex-файл содержимого EEPROM его можно прошить в EEPROM, нажав кнопку «Program» в секции EEPROM.

Таким образом, можно залить в память любую прошивку. Лишь бы она занимала не больше чем 16кб-512б памяти флеша (для контроллера ATmega16). 512байт занимает загрузчик.

На будущее -- я себе AVRProg забил на хоткей внутри системы, это обычный экзешник, лежит в дебрях AVR Studio.

Давайте зальем демопрошивку из третьей части документации.

Запусти снова AVRProg (Не забывайте, что запускать его надо сразу после сброса, пока горит зеленая лампочка загрузчика) и выбирайте **Demo16-RTOS.hex** из третьего архива.

Жмем Program -- пробежит парочка прогрессбаров и в конце **AVRProg** скажет, что все сделано.

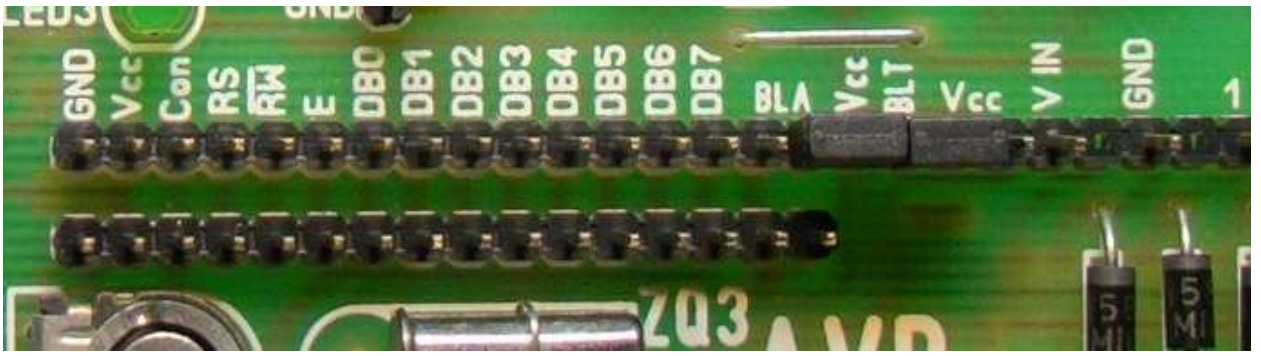
Закройте **AVRProg** и нажмите на **RESET** и только после этого программа стартанет.

Загорится зеленый диод загрузчика, через две-три секунды он погаснет и свежезалитая программа приступит к работе.

Вот только что-то работы не видно? Оно и логично! Все действие только что залитой программы происходит на LCD дисплее, а его то мы и не подключили.

Выключим рубильник Main Power.

Берите дисплей и устанавливай его на линейку расширения. Цоколевка разъема совпадает с обозначениями дисплея. ВLA это анод светодиодов подсветки. Накинув два джампера, как показано на рисунке, мы включим подсветку. Дисплей одевается на нижнюю линейку разъема. К верхней же подключаются проводки соединительные.



Рекомендую сразу же переменный резистор контраста выставить почти в крайне левое положение.



Крутить его удобно с обратной стороны тонкой отверткой. Для этого там есть отверстие в плате.

Главное помнить, что если выкрутить сильно вправо – не будет видно на экране вообще ничего. Если сильно влево – будут только черные квадраты. В общем, если дисплей не показывает того что должен, то надо проверить контрастность.

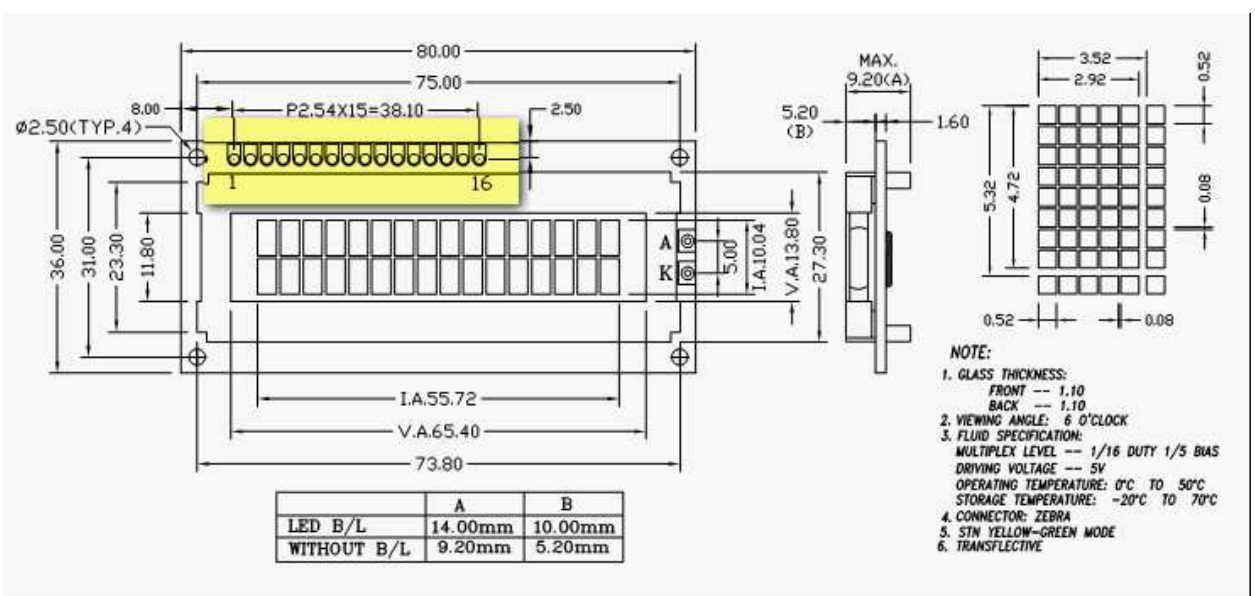
Многие при освоении этих дисплеев тратят прорву времени отлаживая код, а надо то было всего лишь отверточкой покрутить :)

Оптимальный контраст такой, когда при включении, еще до инициализации, на экране виден один ряд серых квадратов.

Теперь потребуется проводок на семь жил.



В нем будет четыре линии данных и три линии управления. 10см длины вполне хватит. Подключается дисплей на PORTB (т.к. в данный момент он не занят ничем серьезным. Порт А обслуживает АЦП, нежелательно на него что-либо вешать в этот момент (снижается точность работы АЦП). Порт С загружен на JTAG, Порт D общается с терминалом. Вот и остается только PORT B в котором мы можем занять сразу много линий. Цоколевка дисплея простая.



## ② MECHANICAL DATA

ITEM	SPECIFICATION	UNIT
Module Size (W X H X T)	80.0 x 36.0 x 9.2	mm
Viewing Area (W X H)	65.4 x 13.8	mm
Number of Character	16 Character x 2 Line	Char
Dot Pitch (W X H)	0.6 x 0.6	mm
Dot Size (W X H)	0.52 x 0.52	mm
Character Pitch (W X H)	3.52 x 5.32	mm
Character Size (W X H)	2.92 x 4.72	mm

## ④ BLOCK DIAGRAM

## ③ PIN CONFIGURATION

ITEM	SYMBOL	LEVEL	DESCRIPTION
1	VSS	L	Power GND
2	VDD	H	Power Positive
3	VEE	L	Voltage For LCD Panel Drive
4	RS	H/L	Register Select
5	R/W	H/L	Read/Write Select
6	E	H/L	Read/Write Enable
7	DB0	H/L	Data Bus
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	A	H	LED BACKLIGHT +
16	K	L	LED BACKLIGHT -

Если смотреть на него сверху, с лица, то нумерация слева направо. Первые три вывода - это питание и контраст. Они уже подключены в колодке. Последние два - питание подсветки. Все что между ними, штыри с 4 по 14 это наши управляющие сигналы. Подключаем шлейфиком с портами микроконтроллера в таком порядке (в скобках номер вывода дисплея):

PB0 - E (6)  
PB1 - RW (5)  
PB2 - RS (4)

PB4 - DB4 (11)  
PB5 - DB5 (12)  
PB6 - DB6 (13)  
PB7 - DB7 (14)

Или посмотрите на рисунке, благодаря цветному шлейфу понятно что куда. Потом шлейф можно уложить аккуратно вдоль МК, чтобы не мешался.

Дисплей подключается по упрощенной схеме, когда на шину данных отводится четыре линии и три на управление. Этот режим сложнее в инициализации, но легче в подключении. Также дисплей может работать в режиме восьми битного порта, тогда потребуется 8 данных и три бита управления.

Подключили! Теперь включаем плату. На дисплее вначале должны сразу зажечься темные квадраты, а потом, спустя пару секунд, пойти надписи.

Вначале будет что то вроде: "АЦП Канал 0", а спустя пару секунд приветствие "Pinboard v1.0".

Если так, то поздравляю. Дисплей уже подключен! Можете попробовать покрутить потенциометр, при этом на дисплее будут показания АЦП. А если сейчас запустить терминалку и подключиться к порту на скорости 9600 то в терминал также будут сыпаться показания с АЦП, но вот если самому туда написать, то это вылезет уже на экран демоплаты.

Осталось подключить кнопки. Делаем еще один шнурок. Вот такой:



И Соединяем линии PD3...PD6 с выводами кнопок ABCD. Аналогично мы делали для версии 1.0.

Теперь, при нажатии кнопок на экране будет высвечиваться имя нажатой кнопки. Все, базовый комплект подключен, дальше можно уже развлекаться самостоятельно.

Попробуйте скомпилировать и загрузить в контроллер программы из учебного курса по Си или ассемблеру, что выложен у меня в разделе AVR. Учебный Курс. На сайте [easyelectronics.ru](http://easyelectronics.ru)

В следующей части документации я расскажу как пользоваться встроенным программатором FTVB и JTAG'ом.

**Да, если решите самостоятельно освоить JTAG то никаких проблем, но учитывайте тот факт, что при подсоединении JTAG к контроллеру в режиме эмуляции из памяти будет вынесен бутлоадер и его придется накатывать заново.**

Так что я предупредил. Впрочем, если затереть бут то ничего страшного не произойдет -- обратно его вернуть это секундная операция, производимая средствами самой платы. Об этом в части про самопрошивку и использование FTVB

**Думаю у вас возникнут вопросы, постараюсь их предугадать и вот небольшой FAQ:**

**В:** Почему подключаемся именно к этим портам? Это какие то специализированные порты?

**О:** Нет, просто я так захотел и мне так было удобно. Прелесть платы в том, что тут нет четкой привязки что к чему подключать. Можно взять и перекинуть дисплей на порт А, например. Но разумеется надо будет внести изменения в программу.

**В:** А почему все проводками? Это что их каждый раз делать? Почему нельзя было сразу плату развести чтобы все было уже подключено.

**О:** Дело в том, что я хотел сделать полностью универсальную плату, без каких либо привязок на аппаратном уровне. Сделать готовые соединения всего со всем не представляется возможным. Поэтому весь основной межблочный монтаж делается навесной лапшой. Однако в этом нет ничего страшного, соединений требуется все же не так много, вряд ли тебе потребуется больше 20-30 жил провода, а это работы на пол часа-

час по заготовке соединительной лапши. Да и монтажно-паяльный опыт никому не будет лишним.

В: Все подключил правильно, но на дисплее так ничего и не зажглось.

О: Скорей всего стоит покрутить ручку контраста яркости дисплея. Также стоит проверить терминалку. Если дисплей подключен неверно, то программа застрянет на инициализации дисплея, а значит в терминалку поток данных с АЦП сыпаться не будет.

В: Не хочу подключать питание от USB. Как подключиться от сетевого адаптера?

О: Без проблем, выставить джампер выбора питания в положение ADAPTER и воткни шнур блока питания в гнездо.

В: А что с Fuse битами делать? Я слышал их надо зашивать программатором.

О: Сейчас прошивка ведется через бутлоадер. Им нельзя изменить Fuse биты. Но вы уверены, что вам сейчас это нужно? Если так сильно хочется, то можно подключить к контроллеру выводы FTBB и через программу avrdude выставить любое значение FUSE бит. Главное осторожней, т.к. можно заблокировать микроконтроллер.

В: Я прошил плату обычным ISP программатором, а потом попытался прошить через загрузчик, но у меня ничего не вышло.

О: При прошивке через ISP если не принять дополнительных мер (включение загрузчика в прошивку) то загрузчик затирается и его надо залить заново. Файл загрузчика есть во втором архиве документации.

Если возникнут вопросы, то пишите мне на [dihalt@dihalt.ru](mailto:dihalt@dihalt.ru)