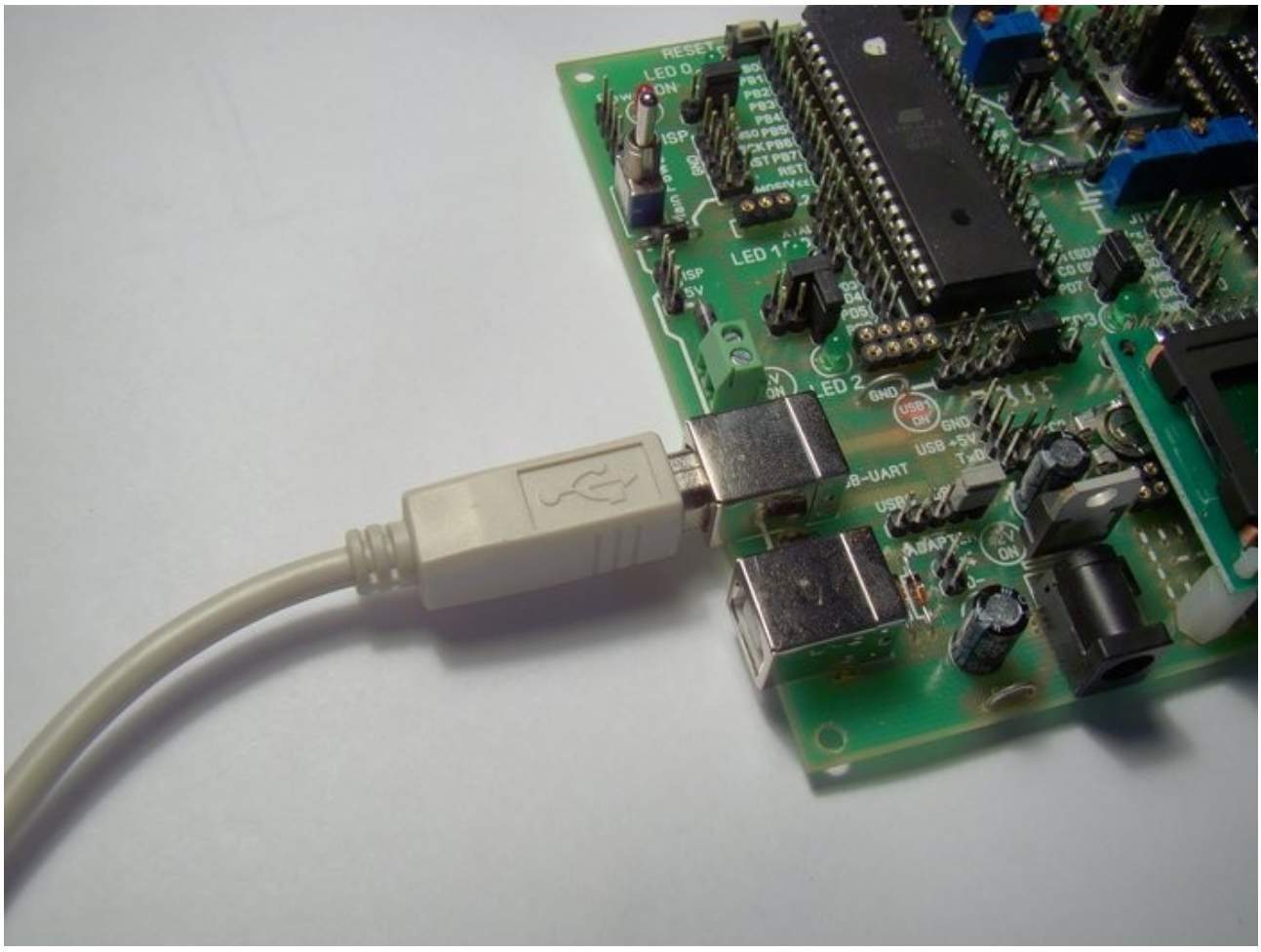
Pinboard on Linux

Автор: Valber
Версия: 0.1

Это краткая инструкция по использованию отладочной платы Pinboard на linux. В данной статье используется софт предлагаемый большинством дистрибутивов, такой как AVRDUDE,gcc-avr. Это пока небольшая инструкция и здесь описана малость (прошивка через bootloader), постараюсь дополнить её, работая с Pinboard

Используется отладочная плата: Pinboard v1.1

Подключение платы.



Вкратце необходимые установки на плате

1).**Voltage Selection** установлен в 5V
2). Переключатели **UART 1,2** установлены ON.
3). Оба переключателя **JTAG 1,2** установлены в OFF.

4).Перемычка питания установлена в положение **USBH**

Для начало необходимо понять установлен ли в вашей системе драйвер FTDI адаптера. Возьмите USB шнур и подключите его к компьютеру, а второй конец к **USB-1**(USB SOFT) на плате Pinboard. Нажмите кнопку **MainPower** Откройте терминал и наберите:

```
ls /dev
```

Если среди всего что вам покажет терминал, есть **ttyUSB**0 то это означает что в вашей системе установлена поддержка FTDI.

Включение поддержки FTDI в ядре

!!!Здесь описаны действия в дистрибутиве Gentoo в чем-то они могут сходится, в чем-то отличаться, если хотите что-то добавить/сделать поправку пишите!

Большинство дистрибутивов распространяются с уже собранным «универсальным» ядром. Если вам необходимо включить поддержку FTDI в ядре(например у вас дистрибутив Gentoo). перейдите в папку с исходными кодами ядра.

Например:

```
cd /etc/source/linux-....папка с ядром
make menuconfig
```

в меню есть поиск (клавиша "/") вы можете найти FTDI драйвер с помощью него. Например у меня в ядре надо было включить.

→Device Drivers
→ USB supports
→ USB serial computer support
→ FTDI single port serial driver

После проделанных изменений сохраните новую конфигурацию.

```
make && make modules
```

Посмотрите куда будет сохранен образ ядра bzImage примонтируйте загрузочный (отдел если надо)

```
mount /boot
cp /адрес где расположено ядро/ bzImage /boot/название ядра
```

под названием ядра имеется ввиду ваше название ядра на которое вы будете ссылаться в grub.conf. Отредактируйте /boot/grub.conf. Отмонтируйте загрузочный раздел:

```
umount /boot
```

перезагрузитесь с новым ядром и снова попробуйте подключить Pinboard как описано выше.

Прошивка через bootloader, стандартным средствами

Буду писать код на C а не на привычном ассемблере, почему? Потому, что синтаксис именно директивы компилятора(вроде .device). Если есть желание можно установить пакет **avra** он позволит скомпилировать ваш ассемблерный файл с синтаксисом Atmel() в hex. GCC(GNU Collection Compilers) тоже может компилировать ассемблер, но там есть отличия и нет(пока не нашел) вменяемой документации и примеров на русском.... хотя и на английском тоже.

Итак вам понадобится установить следующие пакеты(возможно названия отличаются):

1) avrdude прошивающая программа
2) gcc-avr компилятор
3) avr-libc библиотека C-шных функций

Откройте ваш редактор и наберите простейший пример.

```
#define F_CPU 8000000L
#include <avr/io.h>

int main(void)
{
    DDRC=0xFF;   //Порт C на сконфигурирован на выход
    PORTC=0xFF;
    while(1){
    //бесконечный цикл
    }
}
```

Сохраните, например как test.c , затем откройте терминал и перейдите в папку с вашей программой, теперь скомпилируем объектный файл.

```
avr-gcc -mmcu=atmega16 -o test.o test.c
```

или avr-gcc -mmcu=atmega16 -O3 -o test.o test.c , во втором случае мы оптимизируем по размеру выходной файл. Затем транслируем получившийся

файл в формат Intel Hex

```
avr-objcopy -O ihex test.o test.hex
```

Для прошивки вам понадобятся права суперпользователя, ниже напишу как это исправить.

```
sudo su
```

Подключите Pinboard к компьютеру как это было описано выше. Перед прошивкой вы должны нажать клавишу **RESET** рядом с главным микроконтроллером и сразу же начать прошивать(Reset 15с.) :

```
avrdude -p m16 -c avr109 -P /dev/ttyUSB0 -U flash:w:test.hex
```

Тут p — это тип микроконтроллера P — порт c - тип программатора

Подробные инструкции, а также дополнительные опции AVR-GCC ,Документация по AVRdude на русском.

Прошивка Pinboard от пользователя

Например, если устройству соответствует файл /dev/ttyUSB0, выполняем:

```
udevadm info -q path -n /dev/ttyUSB0
```

в результате получаем что-то вроде

```
/devices/pci0000:00/0000:00:1d.3/usb5/5-1/5-1:1.0/ttyUSB0/tty/ttyUSB0
```

0000:00:1d.3 - это и есть **ID** нашего устройства. Соответственно в файл */etc/udev/rules.d/95-avrdude.rules* записываем правило(!В одну строчку!!):

```
KERNEL=="ttyUSB0", ATTRS{serial}=="0000:00:1d.3",  , GROUP="пользователь", MODE="0666"
```

Теперь снова включим и выключим Pinboard. Теперь можно прошивать с пользовательского терминала, а также пользоваться встроенной в Eclipse кнопкой прошивки или графическими оболочками AVRdude!

Вот ресурсы которыми я руководствовался для написания этого раздела: Подробное руководство по udev на русском. Решение с поиском идентификаторов.

Позже напишу более подробное руководство по использованию Eclipse, а пока оно не сильно отличается от написанных AVR+Eclipse Работа с Arduino