

ельское устройство на основе

PICkit2.

Программатор PICKit 2 можно использовать как средство разработки, отладки и знакомства с USB устройствами. Простой пример: вам хочется освоить USB интерфейс. В «стандартном» случае вам нужно иметь программатор и собственно плату с установленным USB контроллером. Если у вас есть программатор PICkit-2 то можно поступить гораздо проще! Сердцем программатора PICkit 2 является USB-контроллер Microchip PIC18F2550. На фабрике в PICkit 2 прошит загрузчик (бутлоадер – bootloader) и прошивка программатора. Бутлоадер необходим для возможности обновления прошивки программатора при выходе новых версий PICkit2. Можно воспользоваться данной возможностью для своих целей, а именно для изучения интерфейса USB. В этом случае все что вам понадобится это PICkit 2. Как заменить программу в PICkit 2.

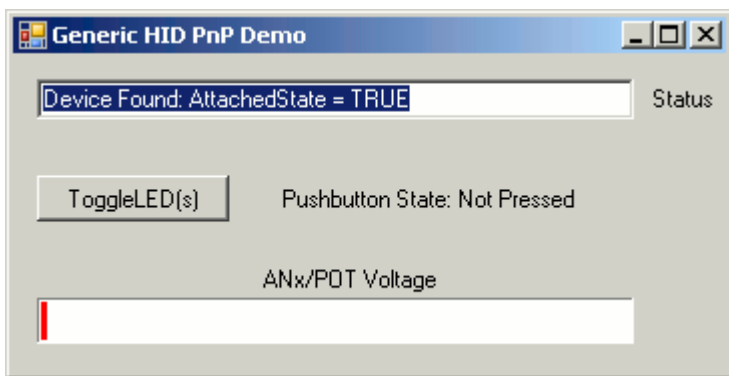
Компания Microchip для поддержки разработок с применением USB PIC контроллеров предлагает пакет библиотек и примеров MCHPFSUSB Framework. На основе готовых примеров создадим прошивку USB HID пользовательского устройства на основе PICkit 2.

Для работы понадобится USB программатор PICkit 2, среда разработки MPLAB IDE, Си компилятор MPLAB C-18 (можно скачать бесплатную студенческую версию с сайта www.microchip.com/C18 [<http://www.microchip.com/C18>]), последняя версия MCHPFSUSB Framework v2.3 (www.microchip.com/USB) [<http://www.microchip.com/USB>] и установленный пакет Visual C++ 2008 Express Edition.

Аналогично [примеру портирования HID клавиатуры](#), модифицируем готовый пример пользовательского HID устройства (папка [USB Device - HID - Custom Demos](#)) из пакета MCHPFSUSB Framework v2.3.

Пример пользовательского USB HID класса показывает как создать устройство отличное от стандартного HID устройства и написать приложение для PC на Visual C++ 2008 Express Edition для обеспечения двунаправленной связи с микроконтроллером.

При запуске приложения для ПК (программа HID PnP Demo.exe), и подключении демонстрационной платы с прошивкой пользовательского USB устройства, мы можем получать данные с движка потенциометра и переключать светодиоды на демонстрационной плате.



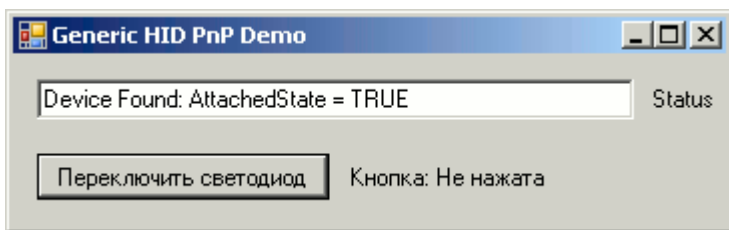
Примечание 1. Для запуска программы HID PnP Demo.exe необходим установленный .NET или Visual C++ 2008 Express Edition (бесплатная программа, доступная с www.microsoft.com [<http://www.microsoft.com>]).

Для портирования примера на PICkit2 нужно:

- изменить файл скрипта линкера
- переопределить перенаправление векторов сброса и прерываний
- переопределить вывод порта управления светодиодом PICkit2 и вход кнопки

Программа HID PnP Demo.exe в первоначальном виде нам тоже не очень нужна, т.к. на корпусе PICkit2 нет того потенциометра который можно покрутить и увидеть результат на экране компьютера. Поэтому удалим из программы все что связано с получением данных от потенциометра.

Модифицированная программа HID PnP Demo.exe теперь будет выглядеть примерно так:



Теперь при нажатии кнопки на PICkit2 мы увидим соответствующее сообщение в окне программы, а при нажатии на кнопку в программе HID PnP Demo.exe переключим состояние светодиода Busy программатора PICkit2.

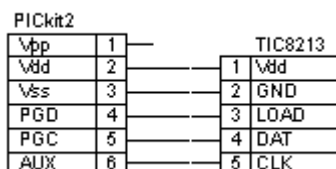
Таким образом мы наладили двухсторонний обмен сообщениями между компьютером и USB устройством.

В данном примере мы задействовали кнопку и светодиод программатора PICkit2. Из доступных ресурсов остается еще:

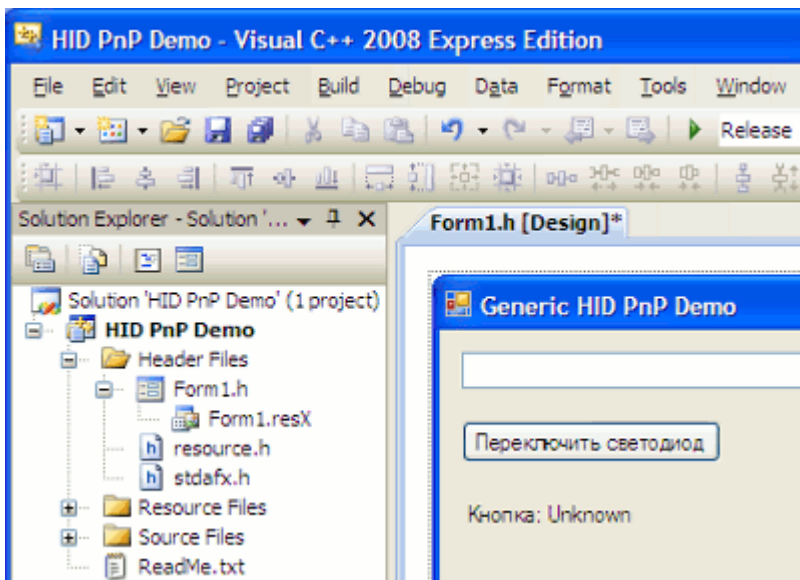
- 3 линии дискретного ввода-вывода
- регулируемый источник напряжения

Модифицируем программу PICkit2 для проверки работоспособности и контрастности ЖК COG (Chip-On-Glass) индикаторов типа TIC8213 и подобных (COG индикаторы [http://www.gamma.spb.ru/products.info.php?c=652&s=459&i=652]), т.е. будем выводить данные на ЖКИ и управлять напряжением питания индикатора.

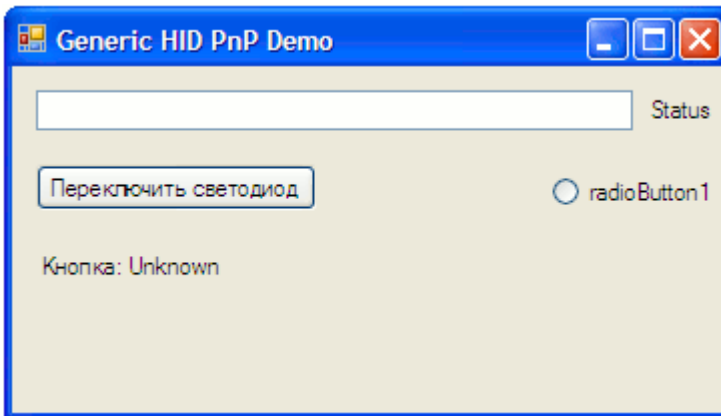
Схема подключения индикатора TIC8213 к PICkit2:



Итак, запускаем файл проекта **HID PnP Demo.sln** в Visual C++ 2008 Express Edition. Открываем файл Form1.h

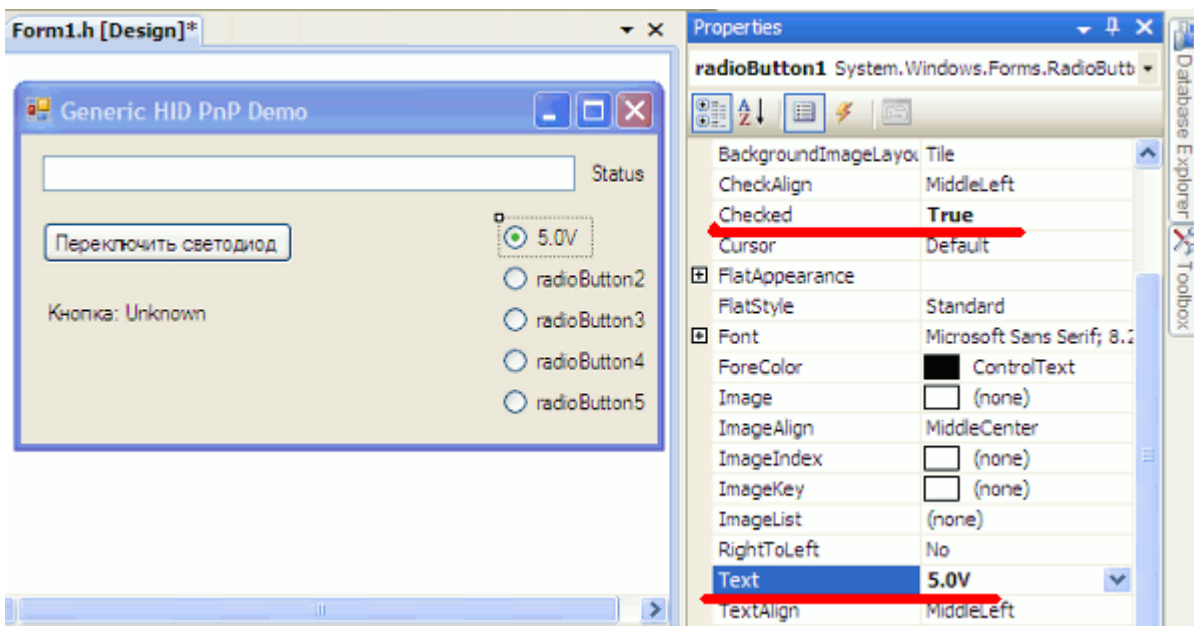


Из боковой панели ToolBox перетаскиваем в форму программы элемент **radioButton**



Всего нужно добавить 5 элементов **radioButton**

Для первой кнопки задаем отображаемый текст **5.0V** и состояние **Checked** → **True**



Для остальных кнопок имена **3.3V**, **2.5V**, **2.0V** и **1.8V**, состояние **Checked** → **False**

В окне редактирования формы **Form1.h** щелкаем правой кнопкой и выбираем **View Code**. В коде программы необходимо добавить обработку кнопок и передачу сообщений в PICkit2 через USB.

```

OUTBuffer[0] = 0; //первый байт "Report ID", не пересылается через USB. всегда = 0.
OUTBuffer[1] = 0x40; //0x40 команда изменения напряжения питания

if(radioButton1->Checked) {
    OUTBuffer[2] = 0x01;
} else if(radioButton2->Checked) {
    OUTBuffer[2] = 0x02;
} else if(radioButton3->Checked) {
    OUTBuffer[2] = 0x03;
} else if(radioButton4->Checked) {
    OUTBuffer[2] = 0x04;
} else if(radioButton5->Checked) {
    OUTBuffer[2] = 0x05;
}
for(i = 3; i <65; i++) { //This loop is not strictly necessary. Simply initializes unused bytes to
    OUTBuffer[i] = 0xFF; //0xFF for lower EMI and power consumption when driving the USB cable.
}
//передаем пакет данных через USB в ПО микроконтроллера
WriteFile(WriteHandleToUSBDevice, &OUTBuffer, 65, &BytesWritten, 0);

```

Теперь можно скомпилировать программу для ПК.

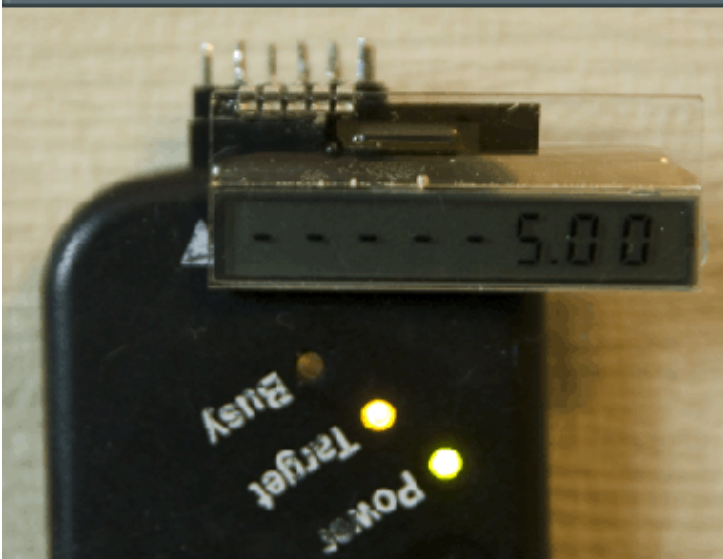
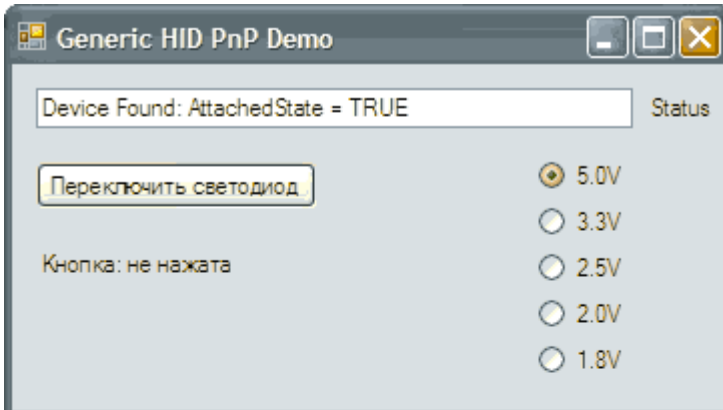
Далее необходимо модифицировать программу для PICkit2: добавить обработку команды смены напряжения питания. В среде разработки MPLAB IDE (проект USB Device - HID - Simple Custom Demo - C18 - PK2.mcp), в файле main.c в обработке команд добавляем следующие строки кода:

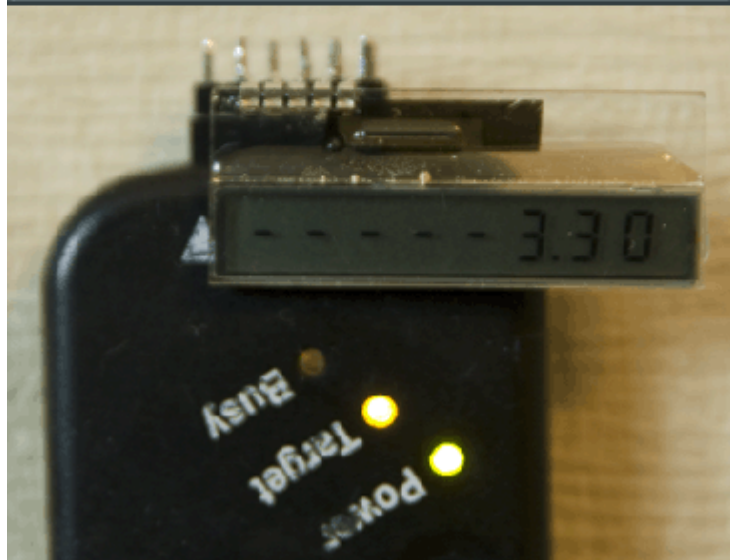
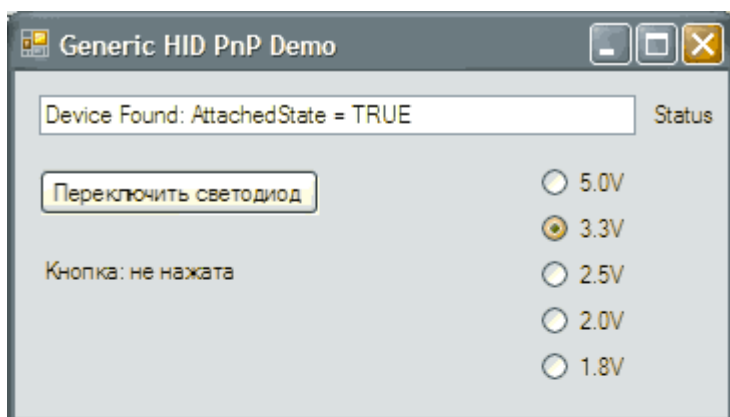
```

case 0x40: // set Vdd
{
  if(ReceivedDataBuffer[1] == 1) { // 5V
    CCPR1L = Vdd_5V0; // PWM duty cycle
  } else if(ReceivedDataBuffer[1] == 2) { // 3.3V
    CCPR1L = Vdd_3V3; // PWM duty cycle
  } else if(ReceivedDataBuffer[1] == 3) { // 2.5V
    CCPR1L = Vdd_2V5; // PWM duty cycle
  } else if(ReceivedDataBuffer[1] == 4) { // 2.0V
    CCPR1L = Vdd_2V0; // PWM duty cycle
  } else if(ReceivedDataBuffer[1] == 5) { // 1.8V
    CCPR1L = Vdd_1V8; // PWM duty cycle
  }
}
break;

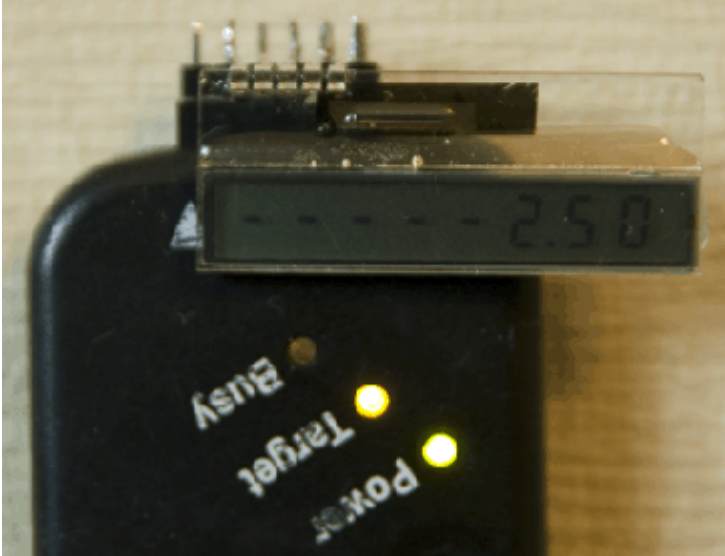
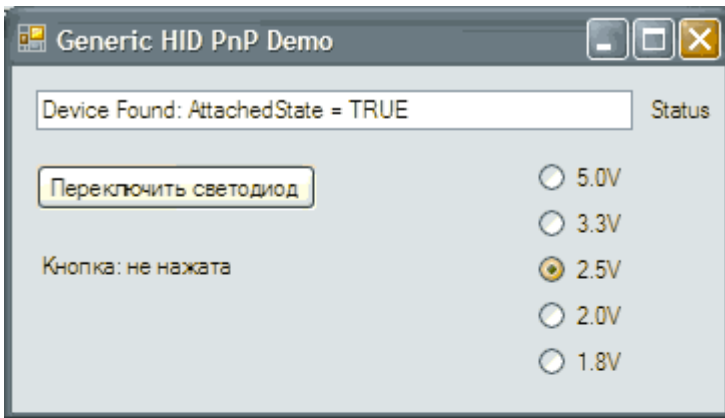
```

Все. Можно скомпилировать программу и прошить ее в PICkit2.

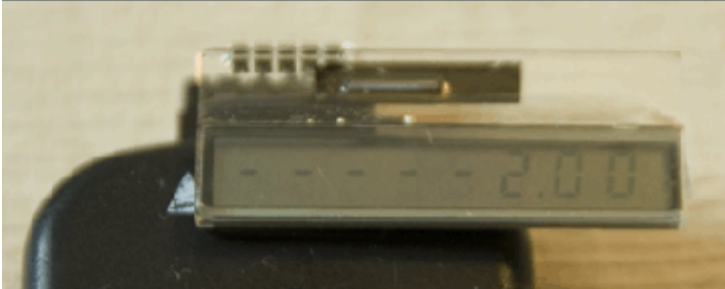
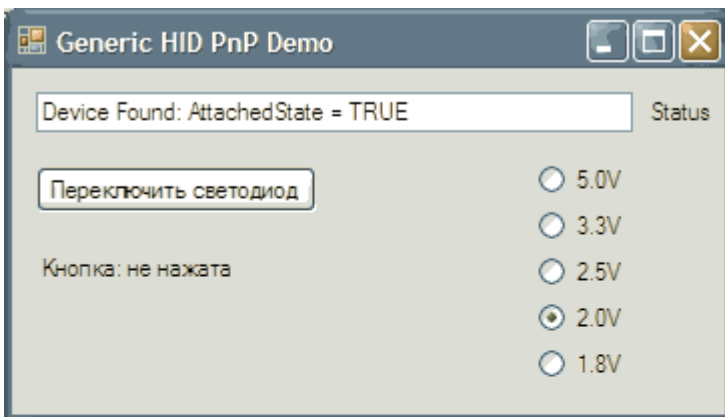


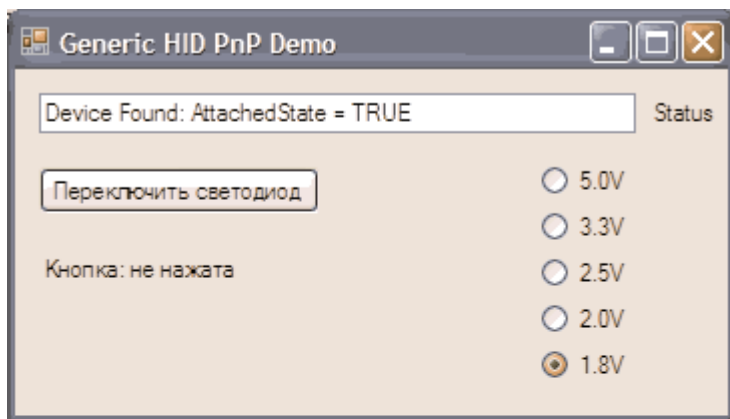


Конкретный испытываемый индикатор предназначен для работы в схемах с напряжением питания 5В, но контрастность не изменяется при снижении напряжения питания до 3.3В



Если прибор должен работать при напряжениях менее 3.3В, то нужен ЖКИ, оптимизированный на более низкое напряжение. В настоящий момент поставляются индикаторы с низковольтными кристаллами с пороговым напряжением 1.8В (низковольтные ЖКИ [<http://www.gamma.spb.ru/products.php?s=459>]).





Исходные коды программы для PICkit2

Исходные коды программы для ПК (под Visual C++ 2008 Express Edition).