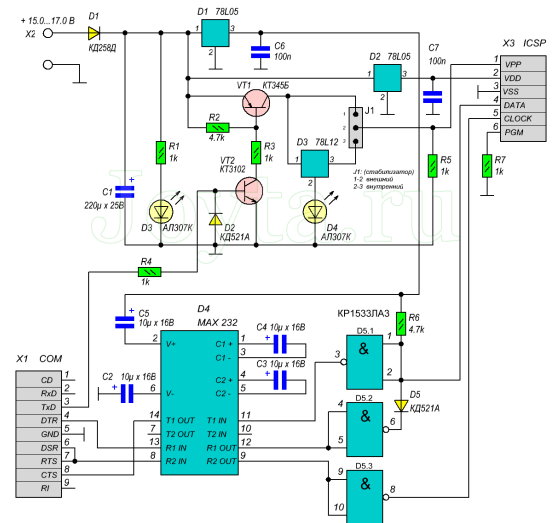


РІС'ами, я такой собрал. Работает хорошо. Только при настройке пришлось повозиться. Я ставил детали которые нашёл:

- MAX232 заменил на ST232, отличие - конденсаторы C2 - C5 нужно ставить на 0,1 uF.

- 1533ЛАЗ заменил на 74НС00 - КМОП серия с питанием от 5 Вольт. Вот тут то и началось! Из за того что переключение в уровень лог.1 у КМОП серии начинается значительно ниже чем у ТТЛ логики - 74НС00 наотрез отказывался переключаться!

Выход был найден - диод КД521А пришлось заменить на диод Шоттки BAT54 имеющим меньшее падение напряжения.



Однако городить такое навороченное устройство для одноразового применения слишком невыгодно.

Всё вышеперечисленное касается только тех пользователей у которых есть доступ к старым компьютерам, у которых еще присутствуют порты COM и LPT.

А как же быть тем кто обновил свой компьютер и из всех интерфейсных разъёмов присутствуют USB?

К сожалению тут придётся городить сложные переходники USB -> UART.

Один из вариантов - собрать на специализированной микросхеме **FT232** или **CP2102**.

Последнюю я купил по случаю, она дешевле чем FT232, требует минимум обвязки, но имеет несколько неудобный корпус для ручной пайки типа **QFN**.

Позже постараюсь собрать на ней ещё один вариант программатора. Ходят слухи что в качестве программатора эта микруха работает очень медленно. Суть в том, что при программировании идёт обмен очень короткими пакетами. А преобразование пакетов USB в последовательность UART и обратно занимает определённое время.

... продолжение уже пишу!

Последний раз редактировалось Волчара; 03.05.2013 в 14:45..



сказали "Спасибо" Волчара

[Mantrid \(03.05.2013\)](#)

03.05.2013, 15:54

#5

Волчара
Общительный



СВОЙ ВАРИАНТ JDM-ОБРУБКА ДЛЯ ПРОГРАММИРОВАНИЯ PIC18F2550

Посмотрим внимательно на вариант LV-программатора работающего от COM-порта. Да, та самая что мне не понравилась изначально. Но у неё есть парочка замечательных свойств. Во первых, схема работает с уровнями -12...+12 Вольт. Для нашего HV-программатора нужно как раз +12 Вольт!

Во вторых, переделывается эта схема для режима HVP весьма быстро - перекидыванием проводка с ножки 26 (PGM) микроконтроллера на ножку 1 (MCLR).

Кстати, питание основной платы будет осуществляться через разъём USB, что несколько упрощает и без того примитивный программатор.

В данном случае **ОБЯЗАТЕЛЬНО** соединить землю программатора и землю платы PICkit 2, это выводы 5 COM-порта -> ножка 3 разъёма ICSP -> выводы 8 и 19 микроконтроллера! Иначе рискуете что-нибудь спалить!

Давайте подумаем - уж коли на выводах COM-порта присутствуют уровни +12 Вольт нам нужно их уменьшить до уровней безопасных для МК. Это примерно 5 Вольт! Как это сделать? Да очень просто! Необходимо лишь защитить линии **стабилитронами**!

Итак, ставим стабилитроны на линии **PGC** и **PGD катодами к линиям, анодами на землю!** В данном случае использовать **двуханодные стабилитроны не рекомендую** и вот почему: уровни лог.0 на выходах COM-порта составляют **-12 Вольт**. В случае применения **двуханодных стабилитронов** могут выгореть защитные диоды внутри МК - напряжение в случае подачи лог.0 на выводе МК будет **-5 Вольт** вместо положенных ~ 0 Вольт! Применение обычных стабилитронов спасает положение - **в прямом направлении они работают как диоды** и падение напряжения составит всего $\sim -0.5...-0.7$ Вольт. Таким образом применив стабилитроны мы сразу "ликвидируем двух ушасть"  - ограничиваем сигналы 5-ю вольтами и защищаем входы МК от большого отрицательного напряжения.

Остался вывод COM-порта переводящий МК в режим HV-программирования. Это вывод 3 (**TxD**). Его также необходимо слегка кастрить - установим диод 1N4148 (можно любой подходящий) **катодом к линии, анодом к GND**. Таким образом отсекается отрицательная составляющая на входе Vpp микроконтроллера.

Обратите внимание на номиналы резисторов в JDM-обрубке: $R1, R2, R4 = 10 \text{ кОм}$, $R3 = 100 \text{ Ом}$.

Минимальная обвязка на плате PICkit 2 состоит из:

- разъёма USB (у меня он miniUSB-B);
- конденсатора по питанию C2;
- конденсатора C1 - обеспечивающего работу внутреннего регулятора 3,3 Вольт для встроенного модуля USB;
- кварцевого резонатора на 20 МГц с конденсаторами C3, C4;
- и кнопки PROGRAM/BOOT с R5, R6, C5.

Пожалуй это всё, что требуется для успешного определения **прошито**(!) устройства при подключении к USB-порту компьютера.

Если Вы решили собирать PICkit 2, то лучше сначала смонтировать сам МК с минимально необходимой обвязкой, а **остальное ставить по мере налаживания, по блокам**, тем самым исключая сразу возможные замыкания в неизвестном месте и выход из строя компонентов. Разумеется это противоречит всем монтажным законам, но монтажники на предприятиях собирают уже **готовое и испытанное серийное изделие**, а не опытный образец в одном экземпляре. Тем более в условиях производства всегда есть специалисты, отыскивающие неисправности в готовых платах, приходящих с участка монтажа. А Вам же придётся делать это самостоятельно.



Изображения

 PICkit2_HVP.jpg (74.0 Кб, 0 просмотров)

Последний раз редактировалось Волчара; 05.05.2013 в 21:23..



03.05.2013, 17:29

#6

Волчара
Общительный



ВЫБИРАЕМ ОБОЛОЧКУ ДЛЯ JDM-ПРОГРАММАТОРА

Допустим JDM-огрызок мы сделали. Возникает закономерный вопрос - железо то есть, а на компьютере чем записывать? Оказывается существуют вполне распространённые оболочки для работы с такими устройствами. Рассмотрим их.

WinPIC800:

Ссылка на сайт: WinPIC800

Последняя версия на данный момент - 3.64H.

Скачиваете архив, устанавливаете, затем инсталлятор установит дрова. Устанавливать лучше не в системный раздел C, а куданибудь на диск D, **пути не должны содержать кириллицу!** Тем самым Вы избежите ошибок при открытии программы

типа "Не найден WP800IO.SYS" особенно на Windows 7, даже с отключенной UAC.

IC-Prog:

Ссылка на сайт: [IC-Prog](#)

Последняя версия на данный момент - 1.06C. При запуске из под учётной записи пользователя, без прав администратора, могут возникнуть проблемы с доступом к портам. Возможно опять поможет установка в директорию диска D без кириллицы.

А теперь самая замечательная программка! **PICPGM:**

Ссылка на сайт: [PICPGM](#)

Последняя версия на данный момент - 1.78. Особенность её в том что она может автоматически определить и тип подключенного микроконтроллера и тип программатора, правда эта функция не всегда работает как положено.

К слову говоря им я и прошивал свой PIC18F2550.

Рассмотрим необходимые настройки для работы с нашим JDM-обрезом: Шансы, что наши "огрызок" и PIC подцепятся "автоматом" невелики, поэтому выставяем всё в точности как на картинке ниже:



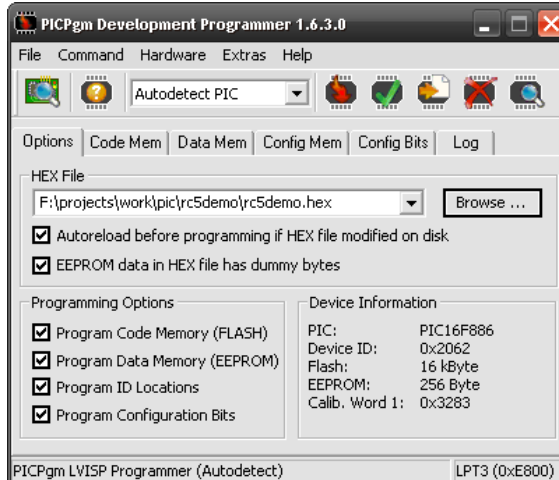
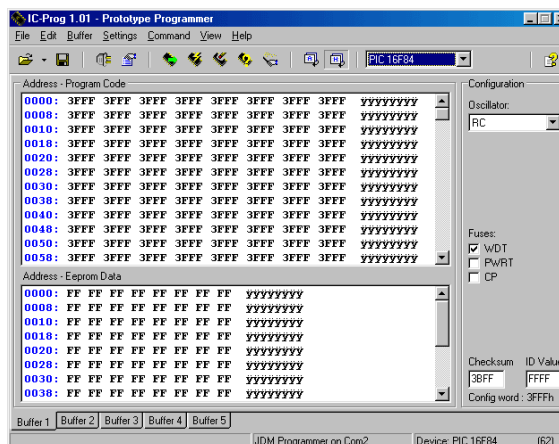
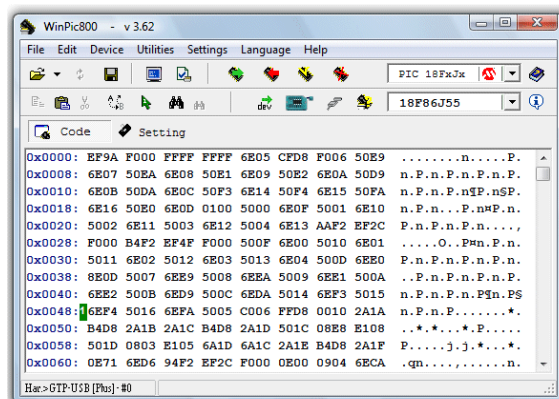
Пройдёмся по ним подробнее:

- в главном окне выбираем тип микроконтроллера - **PIC18F2550**;
- а так же файл прошивки из поста выше - **PK2V023200.hex**;
- ставим все галочки;
- лезем в меню **Hardware Selection/Configuration**;
- в этом окошке выставяем:
 - **JDM-программатор**,
 - **COM1**,
 - проверяем соответствие **pin configuration** схеме нашего "обрубка", на картинке выставлено как должно быть,
 - **Timing delay factor** уменьшаем, скажем до "7",
 - и если всё уже подключено нажимаем **Enable Test**.

Внизу, над кнопочкой "**Ok**" будет указан результат проверки. Если всё в порядке, то жмём **Ok** и кнопку с красным огоньком "Программировать", или выбираем из меню **Command**. Не лишним будет перед этим стереть чип нажав на кнопку с крестиком.

Если же написано **Error** - то отключаем программатор и проверяем соединения и наличие напряжения на МК. Можно нажать **Enable Test** и побаловаться галочками в разделе **Hardware Test** (см. скрин выше), замеряя вольтметром соответствующие напряжения на ICSP или микроконтроллере.

Чуть позже рассмотрим что делать, если МК прошился таки с ошибкой, но не по первому адресу, а где-нибудь в середине или вообще в EEPROM.



Изображения

 [PICPGM.png](#) (434.0 Кб, 0 просмотров)

Последний раз редактировалось Волчара; 03.05.2013 в 22:02..



Эти 2 пользователи сказали
Спасибо Волчара за это
сообщение:

[Mantrid](#) (04.05.2013), [Ясон](#) (03.05.2013)

06.05.2013, 21:01

#7

Волчара 
Общительный



РАЗБИРАЕМСЯ С ОШИБКАМИ ПРИ ПРОГРАММИРОВАНИИ PIC18F2550.

Установим, если ещё не установлена оболочка PICkit2 (с официального сайта мелкочипа, ссылка выше). Тем самым подготовим систему для корректного определения устройства.

Итак, кусок схемы PICkit'a собран, удачно или не совсем удачно прошит , у кого как.

Например, у меня выдал ошибку в EEPROM при верификации (проверке после прошивки, автоматически проводимой софтом программатора). Причём скорость ставил на *slow* - не помогло.

Да и неважно...

И вот почему - в фирменной прошивке PICkit 2 с официального сайта находится конфигурация МК (аналог *фьюзам* в AVR, только здесь она в самой прошивке может находиться). Загрузчик, по сути, представляет собой, на уровне железа, небольшой кусочек кода, располагаемого в памяти программ, конфигурацию и в некоторых случаях определённые данные в EEPROM.

Если у Вас в процессе прошивки выдало ошибку по адресу где то в середине - конце области памяти или в EEPROM, то можно попытаться оживить PICkit2  .

Если у Вас уже припаяны светодиоды BUZY, TARGET, POWER - хорошо, если еще нет, то пришло самое время это сделать...

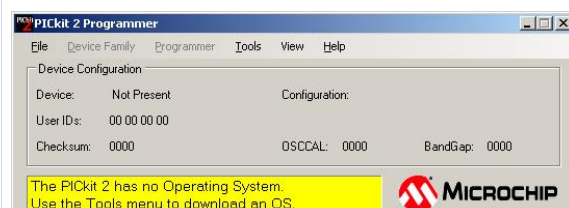
А теперь внимание! Будут следующие пути развития событий:

1. Подключаем PICkit2 к USB порту компьютера. Наблюдаем за его реакцией:

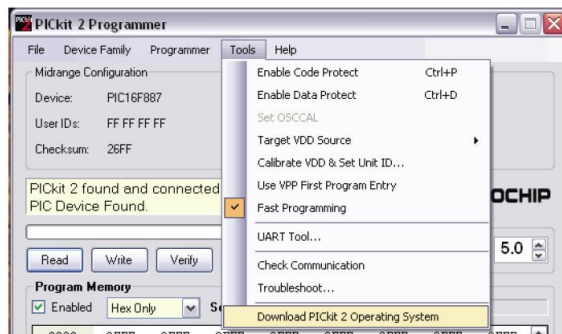
- если всё хорошо - компьютер должен найти новое устройство и установить под него драйвера.
- если устройство определилось, но неправильно - см. п.2.
- если совсем не определилось - надо смотреть на кварцевый резонатор. Как правило с исправным генератором и правильно записанной конфигурацией запускается сразу.

2. Если устройство определяется коряво, то можно попробовать следующее:

- выдернуть USB-кабель;
- запустить оболочку PICkit 2;
- нажать и удерживать кнопку, индикатор **BUZY** должен замигать;
- подключить к USB-порту;
- отпустить кнопку, но **если МК прошивался в LVP режиме - кнопку надо держать на протяжении всего процесса обновления!** В данном случае МК должен находиться в рабочем режиме, а значит при нулевом уровне на выводе PGM.
- если появилась надпись



зайти в меню **Tools**, можно нажать **Check Communication**, а затем **Download PICkit 2 Operating System**;



Должен запуститься процесс программирования.

Почитать про восстановление в принципе Вы можете [здесь](#).

3. Если ничего не привело хотя бы к малейшей работоспособности собранного устройства, то следует попытаться ещё раз прошить JDM- или LV-программатором. Но как правило такие случаи редки.

4. Если **МК читается, не записывается и не стирается**, то вердикт: увы ... 🤖🤖🤖 на свалку - толку от него никакого.

Если всё в порядке, то всё же не помешает обновиться через собственную оболочку. Не зажимая кнопку, выбирайте в меню тот же пункт **"Download PICKit 2 Operating System"**.

Продолжение следует...

Последний раз редактировалось Волчара; 11.05.2013 в 11:37..



сказали "Спасибо" Волчара

[Mantrid \(06.05.2013\)](#)

09.05.2013, 11:30

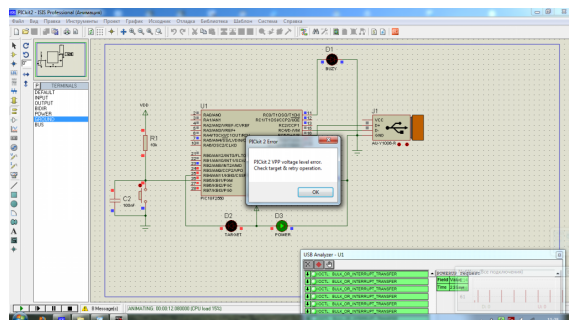
#8

Волчара
Общительный



Всех с праздником! А меня с прошедшим ДР! 🥳🥳🥳

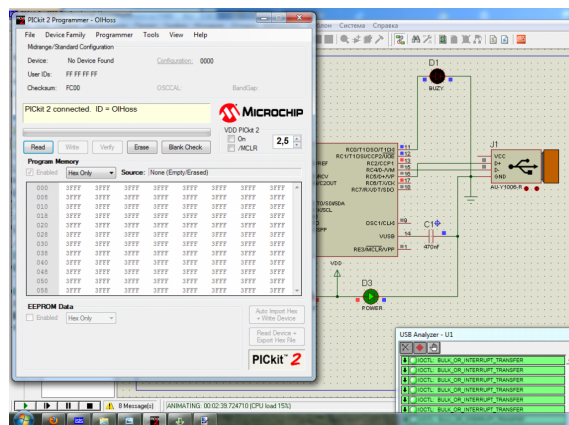
Сегодня праздник, и ради шутки я решил подключить **виртуальный PICKit 2** к оболочке. Смотрите - работает! **И даже ошибку выдаёт при запуске!** 😄



Так как и должно быть с такой урезанной "периферией"! Ведь при запуске софтина пытается провести автотест.

Виртуальный PICKit 2 был собран в среде под общим названием **PROTEUS**. У этой среды есть 2 взаимосвязанных оболочки: схемотехническая **ISIS** и "платоразводочная" **ARES**. Последняя тесно связана с первой.

А вот и софтина запустилась:



Определился!

Последний раз редактировалось Волчара; 09.05.2013 в 11:41..



23.05.2013, 21:37

#9

Волчара Общительный



Итак, мой вариант PICkit 2, собранный временно на картонке. Просьба сильно не глумиться!

Дисплей, что на фотке на отладочной "платке", я сегодня нечаянно испортил - вот не надо было его паять! 😞 Зря не послушал людей, теперь за новым ехать.

[Изображения](#)

[PICkit 2.jpg](#) (57.4 Кб, 0 просмотров)



сказали "Спасибо" Волчара

[Mantrid](#) (23.05.2013)

25.05.2013, 17:54

#10

Волчара Общительный



Продолжим настраивать инструмент.

После удачной прошивки необходимо его **откалибровать**.

Во время этой процедуры PICkit 2 не должен быть подключенным к отлаживаемой/прошиваемой плате! Лезем в меню **Tools/Calibrate VDD & Set Unit ID...** Нажимаем **Clear Calibration** и **Next>**

Опять **Next>**, нахально лезет следующее окошко --->

Подключаем вольтметр к выводу 2 (Vdd) на разъёме **ICSP**. Вписываем измеренное значение в поле и жмём **Calibrate**.

В данном случае у начинающих возникает проблема когда вписанное значение программа не принимает. Обратите внимание на то, что присутствует знак "." или "," - в разных системах по умолчанию может стоять разный. Если выдаст ошибку поменяйте знак!

В следующем окошке Вы можете записать уникальное имя для Вашего программатора. Любое. И нажать **Assign...** оно запишется в EEPROM МК Вашего программатора. Это бывает необходимо в случае если у Вас есть несколько экземпляров программаторов, которые могут быть включены одновременно.

Вот пожалуй и все настройки.

Можете еще устроить проверку с вольтметром. В меню **Tools/Troubleshooting...**

Тестируются **Vdd, Vpp, PGC** и **PGD** - эти можно устанавливать в **1** или **0** и даже выдавать на них пачки импульсов. но тут уже без осциллографа не обойтись. **Разве что судить по показаниям вольтметра - меандр даёт примерно половину напряжения питания.**

Думаю здесь разберётесь уже самостоятельно...

[Изображения](#)

[Калибровка_1.png](#) (118.4 Кб, 0 просмотров)

[Калибровка_2.png](#) (34.5 Кб, 0 просмотров)