**Интерфейс отладки софта по USB для ПК Орион 128**

Для облегчения и существенного ускорения работы по разработке и отладки ПО для ПК Орион-128 разработан интерфейс связи по USB на основе чипа FT245RL с современным компьютером и установленной системой Windows XP - 7 - 8 - 10. Плата интерфейса устанавливается вместо микросхемы последовательного порта 580ВВ55. (с версии 2 модуль USB на работу PIO 580вв55 не влияет, т.е при работе с PIO и USB ничего переставлять не надо )

При подключении к USB модуль интерфейса определяется как последовательный порт СОМ-х.

С другой стороны на ПК Орион 128 интерфейс виден как параллельный порт по двум адресам, F600H - регистр статуса порта ; F601H - регистр приема-передачи, далее адреса чередуются.

Для поддержки интерфейса создан софт отладчика для Windows на основе VBS и загрузчик на ASM для Орион-128 который размещен на ROM-диске.

Работа с отладчиком очень простая: В системе Windows в любом текстовом редакторе пишется текстовый файл с кодом ассемблера и кидается мышкой на скрипт отладчика DEBUG.VBS, далее скрипт делает все автоматически (в зависимости от уставок в шапке скрипта) компилирует код и создает файл bin, lst, bru в папке проекта, затем отправляет файл bru в СОМ порт (и, или в эмулятор). Далее предварительно запущенная программа - загрузчик находящаяся в пк Орион-128 принимает файл и сохраняет по адресу в памяти указанном в шапке bru и передает управление по адресу загруженного софта. В результате получаем отладчик в железе или в эмуляторе (зависит от уставок) практически “ОДНОЙ КНОПКОЙ”

**Настройка:**

Первым делом создаем папку проекта в любом удобном месте(в пути проекта избегайте использование пробелов и спецсимволов таких как ;:@#& и т.д.)

Помещаем в папку проекта файл Debug.vbs и my.asm.

Файл my.asm является шаблоном для последующего изменения его в Ваш код.

В шапке файла скрипта Debug.vbs, прописываются полные пути к файлам компилятора и эмулятора, например:

**PathCoder** = “D:\...\TASM.EXE”

**PathEmul**  = “D:\...\OrionZEm.exe”

эмулятор брал здесь <https://github.com/serge-404/OriZEmu>

**ниже приведен текст автора программы эмулятора:**

| Orion/Z (Orion-128 + Z80-CARD-II, Orion-PRO) emulator, version 1.08  Author: Sergey A. <a-s-m@km.ru>  Copyright (c) 2011 Sergey À.  This program is free software; you can redistribute it and/or modify it in any ways.  This program is distributed "AS IS" in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. | Орион/Z Эмулятор (Orion-128 + Z80-CARD-II, Orion-PRO),  версия 1.08  Автор: Сергей А. <a-s-m@km.ru >  Copyright (c) 2011 Сергей А.  Эта программа является свободным программным обеспечением; вы можете распространять ее и/или  модифицируйте его любым способом.  Эта программа распространяется "КАК ЕСТЬ" в надежде, что она будет  полезна, но БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ; даже без подразумеваемой гарантии ТОВАРНОЙ ПРИГОДНОСТИ или ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ  цели. |
| --- | --- |

**FileAsm** = "my.asm" - полный путь не обязателен. При запуске скрипта обрабатывается этот файл. Если указать пустую строку “” то скрипт вызывает диалог выбора файла. Запуск компиляции так же можно вызвать перетаскиванием мышкой файла "my.asm" на Debug.vbs.

Если на скрипт кинуть файл с расширением bru, скрипт его просто отправит без компиляции.

**ComSet**  = "COM5:9600,N,8,1" - Настройки СОМ порта.

**CorrByte** = &H00 - Байт которым дописывается файл при включении флага “FileCorrect16”

Далее настраиваем параметры шапки BRU файла: Параметр “**StartAdr = &H0100**” должен совпадать с адресом указанным в файле .asm после  **ORG 0100H**.

**BruHeadName**  В этом параметре указывается имя файла которое состоит из восьми символов. Далее четыре служебных байта. В скрипте используется параметр “**Atr3**” для передачи команды управления действиями загрузчика на пк Орион 128.

**bit(0)** = 0/1 - Разрешение вывода на экран подробностей загрузки: “1” - разрешено;

“0” - запрещено и блокирует действия bit(1) и bit(2)

**bit(1)** = 0/1 - При загрузке выводит на экран первые 16 байт загружаемого файла в символьном и шестнадцатеричном виде, по умолчанию - 0 .

**bit(2)** = 0/1 - “1” вывод листинга дампа памяти загруженного файла, по умолчанию - 0

**bit(3)** = 0/1 - Автозапуск после загрузки: “1” - разрешен; “0” - запрещен

**bit(4)** = 0/1 - Переключатель передачи управления: “1” - загруженная программа

“0” - переход в монитор по адресу F800H.

**Флаги:**

**NotepadFlag** = 0/1: “1” - Запускается notepad.exe c результатами компиляции .lst

**EmulFlag** = 0/1: “1” - После компиляции загружается эмулятор и в него передается вновь созданный файл .bru.

**SendCOMflag** = 0/1 “1” - После компиляции файл .bru отправляется в СОМ порт

**FileCorrect16** = 0/1 “1” - После компиляции перед созданием файла .bru

к файлу .bin добавляются байты до конца строки кратной 16 и изменяется длина файла в шапке, таким образом файл .bru полностью подготовлен для добавления в ROM диск для работы с ORDOS

**GetHeadName** = 0/1 “1” - В шапке bru используется имя файла (NAME $).asm

“0” - Используется имя явно указанное в настройках скрипта

**Отличия версий:**

**V1**: простейший вариант схемы подключения модуля к пк Орион-128 с двумя корпусами дополнительной логики 555ла3 и 555ле1. устанавливается вместо 580вв55.

Драйвер **v1:** С фиксированным адресом F600 - слово состояния, F601 - данные

**V2**: в схему введено дополнительно три корпуса логики для управления триггером переключения режимов работы чтобы обеспечить совместную работу модуля не затрагивая функций работы портов ввода вывода по адресам F400, F500, F600.

Принцип совместной работы модуля USB с портом PIO таков: в микросхеме 580ВВ55 в регистре управляющего слова на бит(7) выставляется “0” микросхема переходит в режим оперирования битами см. **таблица 1**.

В этом режиме имеются три неиспользуемых резервных бита бит(6) бит(5) бит(4) которые и были использованы для переключения режимов работы PIO или USB модуля.

**Бит(6)** 0 = подключается PIO 580вв55. 1 = подключается USB модуль.

**Бит(5)** управление светодиодом “Готовность” 1 = вкл. 0 = выкл.

Драйвер **v2:**  Используются три драйвера каждый скомпилирован на три разных адреса портов F400, F500, F600 соответственно

**V3**: в схеме сделана оптимизация логики элементы 555ле1 которые использовались как инверторы, заменены на 555лн1 в результате появилось два дополнительных инвертора которые были использованы в цепи сброса и цепи защелкивания триггера 555тм2 в более правильный момент импульса.

Драйвер **v3:** В этом драйвере добавлена функция автодетекта, драйвер самостоятельно определяет в какой порт установлен модуль USB. Проверка выполняется за два этапа. На первом этапе в управляющее слово по адресу F403 записывается 0 затем читается и запоминается значение по адресу F400. Далее записывается по адресу F403 01000000b, чем переключается триггер на управление модулем, по адресу F400 читается и сравнивается с сохраненным значением. Если значение не изменилось значит модуль не установлен, переходим к проверке следующего адреса F500. Если же значение изменилось значит модуль возможно установлен и видится слово его состояния переходим ко второму этапу проверки. На втором этапе сравниваются значения адресов F400 и F402 так как в модуле они чередуются, соответственно значения по этим адресам должны быть одинаковые. Поле того как адрес модуля будет определен значения определившихся адресов сохраняются в памяти и запускается основная программа загрузчика.

Таблица 1. Управляющее слово 580вв55 оперирование битами

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| Флаг управления  0 = оперирование битами | Произвольное состояние | | | Выбор бита:  0 0 0 = 0 бит  0 0 1 = 1 бит  0 1 0 = 2 бит  0 1 1 = 3 бит  1 0 0 = 4 бит  1 0 1 = 5 бит  1 1 0 = 6 бит  1 1 1 = 7 бит | | | 1 = установить бит  0 = снять бит |

В этом проекте находится последняя версия софта понимающая все версии плат с модулем.

Модуль с версией 1 устанавливается только в порт с адресом F600.