

Внутрисхемный отладчик MPLAB-ICD2

Внутрисхемный отладчик MPLAB-ICD2 становится самым популярным отладочным средством микроконтроллеров компании Microchip Technology, Inc. Во-первых, это связано с возможностью осуществлять внутрисхемную отладку и программирование всех Flash-контроллеров Microchip, в том числе и 16-разрядных с ядром DSP. Во-вторых, все новые PIC и dsPIC контроллеры Microchip выпускаются со встроенным механизмом ICD (In-Circuit Debugger), а с выходом очередного обновления среды разработки MPLAB IDE программное обеспечение отладчика ICD2 обновляется для поддержки новых контроллеров. Таким образом, приобретая отладчик, вы получаете гарантированную поддержку своих будущих разработок.

Алексей САФРОНОВ
Илья НИКИФОРОВ
safronov@gamma.spb.ru

Введение

С помощью ICD2 отладка программы осуществляется в микроконтроллере, установленном непосредственно в разрабатываемое изделие, и программа выполняется с реальной периферией и сигналами, что обеспечивает учет всех особенностей устройства. Для того чтобы функционировал режим внутрисхемной отладки, в выпускаемые микроконтроллеры встраивают специальный механизм поддержки ICD (In-Circuit Debugger). Для осуществления отладки этот механизм можно включить, а для серийных изделий — выключить в конфигурационном слове микроконтроллера.

Принцип действия механизма следующий: во время работы микроконтроллера при достижении установленной точки останова или при пошаговой отладке обрабатывается немаскируемое прерывание и управление передается подпрограмме отладчика (она незаметно для пользователя дописывается в последние ячейки программной памяти). Эта подпрограмма выполняет функцию передачи через дебаггер в компьютер состояния ячеек памяти микроконтроллера, а также изменяет их состояние и переключает режимы работы по команде с компьютера. В результате пользователь может выполнять программу в реальном времени (программа выполняется в соответствии с тактовой частотой контроллера), осуществлять пошаговую отладку (когда шаг за шагом, по команде с компьютера, выполняется код) и видеть состояние нужных регистров в среде разработки MPLAB IDE как при программировании на языке ассемблера, так и на языке высокого уровня Си. Поддерживается установка точек останова, просмотр и изменение памяти данных ОЗУ и EEPROM.

В режиме отладки становятся недоступными: 1 или 2 уровня стека (в зависимости от се-

мейства отлаживаемого контроллера); порты ввода/вывода RB6 и RB7 (для программирования микроконтроллера и управления режимами отладки); вывод MCLR/Vpp (используется для программирования). Помимо этого, при отладке также становятся недоступной для пользователя часть ячеек программы и регистров ОЗУ, которые резервируются для работы подпрограммы отладчика.

Следует заметить, что ограничения вносятся лишь при включенном режиме отладки. Если же программа очень большая, то можно порекомендовать отлаживать программу частями, а затем отключить режим отладки и полностью запрограммировать используемый микроконтроллер. При этом снимаются все налагаемые MPLAB-ICD2 ограничения, он работает в режиме обычного программатора.

Так как отладчик MPLAB-ICD2 программирует Flash-память программ контроллеров, то с помощью него можно не только отладить программу в разрабатываемом устройстве, но и использовать его в качестве серийного внутрисхемного программатора (в ICD2 встроена защита от перегрузок по току и напряжению, есть диагностические светодиоды контроля состояния). Невысокая стоимость дебаггера делает его весьма привлекательным многофункциональным отладочным средством.

Особенностями дебаггера MPLAB-ICD2 являются:

- быстрая связь с компьютером через USB или RS-232;
- три встроенных светодиода, характеризующих состояние дебаггера;
- встроенная система внутрисхемного программирования отлаживаемого микроконтроллера;
- постоянное обновление программного обеспечения;

- задаваемые точки останова (количество зависит от контроллера);
- работа в пошаговом режиме и режиме реального времени;
- просмотр и модификация содержимого управляющих регистров, Flash-памяти, RAM и EEPROM;
- работа во всем диапазоне частот.

Начало работы

Прежде всего для работы с ICD2 необходимо установить ПО: MPLAB IDE.

Далее выбрать способ соединения ICD2 с ПК: через RS-232 или USB (меню Debugger (Programmer) -> Settings -> Communication). В дальнейшем используйте только выбранное соединение.

При использовании RS-232 никаких дополнительных драйверов не требуется.

Если вы используете USB-соединение, то действуйте согласно инструкциям, расположенным в папке, куда установлен MPLAB IDE: MPLAB IDE \ICD2\Drivers\Ddcd2.htm. Подключать ICD2 к компьютеру по USB можно только после установки драйверов!

Обратите внимание: чтобы избежать повреждения ICD2, соедините ICD2 с ПК через USB-кабель, только после этого можно подключать отлаживаемое устройство и его источник питания! Так же в конце работы USB-кабель отсоединяется в последнюю очередь. А выключение ПК возможно только после отсоединения USB от ICD2.

Возможно несколько вариантов подключения ICD2:

- 1) Использование RS-232 для соединения с ПК, питание на отлаживаемую плату (ток до 200 мА, напряжение 5 В) подается через ICD2.
- 2) Использование RS-232 для соединения с ПК, отлаживаемая плата питается отдельно.

3) Использование USB для соединения с ПК, питание на отлаживаемую плату подается через ICD2.

4) Использование USB для соединения с ПК, отлаживаемая плата питается отдельно.

Для подачи питания на отлаживаемое изделие необходимо установить флажок "Power target circuit from MPLAB ICD2" в меню Debugger(Programmer)->Settings->Power.

Для избежания повреждения выходных каскадов дебаггера необходимо соблюдать следующую последовательность:

1. Подключить ICD2 к компьютеру и выбрать его (Debugger(Programmer)->Select Tool->MPLAB ICD2).

2. Подключить незапитанную отладочную плату к ICD2.

3. Подать питание на плату.

Отключение производить в обратном порядке.

Программирование МК

Для того чтобы запрограммировать МК, необходимо выполнить следующее.

1. Выбрать ICD2 в качестве программатора: Programmer->Select Programmer->MPLAB-ICD2, Programmer->Connect (или кнопку с панели инструментов "Reset and Connect to ICD"). Дождаться в окне Output->MPLAB-ICD2 сообщения "MPLAB-ICD2 Ready".

2. Скомпилировать проект в MPLAB IDE Project->Build All (Ctrl+F10) или кнопкой "Build All" на панели инструментов (перед этим нужно выбрать тип микроконтроллера Configure->Select Device). Дождаться в окне Output->Build сообщения "BUILD SUCCEEDED".

3. Выбрать необходимые опции программирования МК в закладке Programmer->Settings->Program.

4. Установить нужные значения в слове конфигурации МК в меню Configure->Configuration Bits.

5. Programmer->Program или нажать кнопку Program target device с панели инструментов.

Внутрисхемная отладка

Для внутрисхемной отладки необходимо в конфигурационном слове контроллера:

- правильно задать тип генератора в соответствии с используемым в отлаживаемом устройстве генератором (наличие тактового генератора необходимо для отладки устройства);
- выключить сторожевой таймер WDT (WDT — disabled);
- выключить защиту кода (Code Protect — disabled);
- выключить защиту чтения таблиц (Table Read Protect — disabled);
- включить режим отладки (Background Debug — enabled);
- выключить низковольтное программирование (Low Voltage Program — disabled).

Чтобы загрузить в МК программу для отладки, необходимо сделать следующее.

1. Выбрать ICD-2 в качестве отладчика: Debugger->Select Tool->MPLAB-ICD2, Debugger->Connect (или кнопку с панели инструментов "Reset and Connect to ICD").

2. Скомпилировать проект в MPLAB IDE: Project->Build All (Ctrl+F10) или кнопкой "Build All" на панели инструментов (перед этим нужно выбрать тип микроконтроллера Configure->Select Device).

3. Выбрать необходимые опции программирования МК в закладке Debugger->Settings->Program.

4. Установить нужные значения в слове конфигурации МК в меню Configure->Configuration Bits.

5. Debugger->Program или нажать кнопку Program target device с панели инструментов.

Теперь можно приступить к внутрисхемной отладке.

1. Работа МК в реальном времени.

Debugger->Run, F9 или кнопка Run с панели инструментов. МК будет выполнять программу, пока не выполнится одно из условий точек останова либо не будет нажато Debugger->Halt, F5 или кнопка Halt на панели инструментов. Если после остановки снова нажать Run (Animate, Step Into, Step Over), программа продолжит выполнение кода с того места программы, где она находилась в момент остановки. Для сброса программного счетчика на начальный вектор необходимо после команды Halt выполнить Debugger->Reset, F6 или Reset на панели инструментов.

2. Работа МК в режиме Animate.

Debugger->Animate или кнопка Animate с панели инструментов. МК будет автоматически пошагово выполнять программу, пока не выполнится одно из условий точек останова либо не будет нажато Debugger->Halt, F5 или кнопка Halt на панели инструментов. В отличие от режима Run на каждом шагу будет обновляться информация, отображаемая в окнах "Special Function Register" и Watch. Если после остановки снова нажать Run (Animate, Step Into, Step Over), программа продолжит выполнение кода с того места программы, где она находилась в момент остановки. Для сброса программного счетчика на начальный вектор необходимо после команды Halt выполнить Debugger->Reset, F6 или написать Reset на панели инструментов.

3. Работа МК в пошаговом режиме.

МК выполняет один последующий шаг (1 команду) по коду программы (смещается на 1 инструкцию в ассемблерном коде либо на 1 строку Си-кода) по команде Debugger->Step Into, F7 или Step Into на панели инструментов. При этом в том случае, если выполняемой командой является вызов подпрограммы, отладчик «зайдет» в нее.

По команде Step Over, F8 или Step Over на панели инструментов МК выполняет 1 шаг (1 команду) по коду программы (смещается на 1 инструкцию в ассемблерном коде либо на 1 строку Си-кода). В том случае, когда выполняемой командой является вызов подпрограммы, отладчик использует свободную точку останова (если таковая есть) для того, чтобы «перешагнуть» подпрограмму, то есть попасть на адрес, следующий за пропускаемой функцией. Если все точки останова задействованы, то отладчик автоматически переключит режим работы в Animate и остановится, выполнив пропускаемую подпрограмму.

4. Точки останова.

Установить точку останова на нужной строке кода программы можно двойным щелчком левой кнопки мыши, либо установив предварительно курсор мыши на соответствующую строчку. Об установленной точке останова свидетельствует значок ■ слева от строки. Для снятия точки останова выполните двойной щелчок левой кнопкой мыши на строке, в которой она установлена. Таким образом можно устанавливать точки останова в ассемблерных и Си-файлах проекта, а также непосредственно в окне View->Program Memory.

При работе МК в режиме реального времени имеет место такой эффект, как «проскальзывание» (skidding): отладчик останавливается не в точке останова, а спустя 2 команды. Следует обратить внимание, что этот эффект отсутствует при пошаговой отладке, так как в режиме Animate MPLAB знает адрес следующей инструкции до выполнения.

Новые МК имеют не только простейшие точки останова по совпадению программного счетчика с заданным адресом, но также продвинутые (advanced breakpoints) и новые возможности, удобные для отладки.

- Останов по совпадению программного счетчика (Program memory execution).

Меню Debugger->Advanced Breakpoints...->Breakpoints.

Breakpoint Type: Program Memory Execution. Program Memory Address: значение программного счетчика в шестнадцатеричном виде, при достижении которого должен произойти останов.

Pass Count Type:

- 1) Pass counting disabled — останов при первом же достижении заданного адреса;
- 2) Event must occur Passcount times — в Passcount надо задать число от 0 до 255, останов произойдет после того, как заданное число раз значение программного счетчика совпадет с заданным адресом;
- 3) Break occurs Passcount instructions after event — в Passcount надо задать количество инструкций от 0 до 255, которое будет выполнено после совпадения значения программного счетчика с заданным адресом до останова.

- Останов по чтению/записи по заданному адресу в памяти данных.
Меню Debugger->Advanced Breakpoints...->Breakpoints.
Breakpoint Type: X Bus Read/Write.
Data Memory Address: в шестнадцатеричном виде адрес памяти данных, при чтении/записи по которому выполнится останов.
Назначение Pass Count аналогично.
- Останов по чтению/записи по заданному адресу в памяти данных заданного значения.
Меню Debugger->Advanced Breakpoints...->Breakpoints.
Breakpoint Type: X Bus Read/Write Specific Byte/Word.
Data Value: значение в шестнадцатеричном виде соответствующего формата (Byte/Word), при чтении/записи которого по заданному адресу в памяти данных произойдет останов.
Назначение Pass Count и Data Memory Address аналогично.
- Комбинирование точек останова.
Меню Debugger->Advanced Breakpoints...->Breakpoint Combinations.
Задание последовательного выполнения условий для останова: в рабочем поле Breakpoint Sequencing выставьте флажки, задающие последовательность условий точек останова (эти условия (breakpoint 0, breakpoint 1, breakpoint 2, breakpoint 3) задаются на вкладке Breakpoints того же окна), выполнение которых друг за другом в заданном порядке приведет к останову. Наличие верхнего флажка (Breakpoint 1 must occur before breakpoint 0) свидетельствует, что для останова условие breakpoint 1 должно выполниться до breakpoint 0. Аналогично, средний и нижний — breakpoint 2 до breakpoint 1 и break-

point 3 до breakpoint 2 соответственно. Если последовательность условий необходимо изменить, то в рабочем поле Swap Breakpoints нужно в списках Swap Breakpoint и With Breakpoint выбрать те breakpoint'ы, которые требуется поменять местами, и нажать Swap.

Одновременное выполнение нескольких условий приводит к останову. В рабочем поле ANDED Breakpoints поставьте флажки у тех условий, одновременное выполнение которых приведет к останову.

- Останов по срабатыванию сторожевого таймера.
Меню Debugger->Advanced Breakpoints...->Emulator Features->Break On Watchdog Timer.
- Останов по вхождению в Sleep.
Меню Debugger->Advanced Breakpoints...->Emulator Features->Break On SLEEP. МК не входит в Sleep, а программный счетчик принимает значение следующей после «пробуждения» инструкции.
- Измерение временных промежутков посредством Stopwatch.
Меню Debugger->Advanced Breakpoints...->Emulator Features->Enable Stopwatch.
При установленном флажке Halt on Start Condition (Breakpoint 1) при выполнении условия breakpoint 1 программа приостанавливается и стартует Stopwatch. Если флажок не установлен, то Stopwatch стартует без остановки программы. При установленном флажке Halt on Stop Condition (Breakpoint 0) при выполнении условия breakpoint 0 программа останавливается и останавливается Stopwatch. Если флажок не установлен, то Stopwatch останавливается без остановки программы. Флажок Reset Stopwatch on Run сбрасывает Stopwatch в ноль при каждом последующем запуске программы.

Заключение

Итак, внутрисхемный отладчик MPLAB-ICD2 — наиболее дешевое решение для программирования и отладки изделий на базе контроллеров Microchip. ICD2 поддерживается всеми Flash-микроконтроллерами Microchip, а на сайте microchip.com доступны для бесплатного скачивания необходимое ПО (MPLAB IDE) и обновление «прошивки» ICD2. ICD2 имеет ряд полезных и удобных функций для отладки программ: разные режимы работы (отладка в реальном времени и пошаговая отладка), продвинутые точки останова (по совпадению адреса программного счетчика, по совпадению адреса/значения чтения/записи в памяти данных, по вхождению в режим Sleep, по срабатыванию сторожевого таймера), а также их комбинирование, возможность просмотра и модификации содержимого управляющих регистров, Flash, RAM и EEPROM, возможность измерения временных интервалов между точками останова. При всей многофункциональности и универсальности ICD2 прост в применении и не требует никаких дополнительных средств для отладки и программирования PIC-контроллеров. В отличие от эмуляторов внутрисхемный отладчик работает с тем контроллером, который будет установлен в конечном устройстве, то есть отладка происходит с реальным контроллером. Внутрисхемный программатор-отладчик MPLAB-ICD2 является сегодня самым популярным программатором среди разработчиков устройств на базе контроллеров 8- и 16-разрядных микроконтроллеров Microchip и зарекомендовал себя в качестве надежного и удобного отладочного средства. ■