

Средства Разработки Программного Кода PIC Контроллеров

Цель работы: Освоение средств программирования PIC контроллеров

Задача работы: Разработка программ на ассемблере для PIC контроллеров в среде MPLAB IDE, компиляция кода Simulink моделей для PIC контроллеров.

Приборы и принадлежности: Персональный компьютер, демонстрационная версия MPLAB IDE 8.92, Matlab/Simulink.

ВВЕДЕНИЕ

PIC-контроллеры остаются популярными в тех случаях, когда требуется создать недорогую компактную систему с низким энергопотреблением, не предъявляющую высоких требований по ее управлению. Эти контроллеры позволяют заменить аппаратную логику гибкими программными средствами, которые взаимодействуют с внешними устройствами через хорошие порты.

Миниатюрные PIC контроллеры хороши для построения преобразователей интерфейсов последовательной передачи данных, для реализации функций "прием – обработка – передача данных" и несложных регуляторов систем автоматического управления.

Компания Microchip распространяет MPLAB - бесплатную интегрированную среду редактирования и отладки программ, которая записывает бинарные файлы в микроконтроллеры PIC через программаторы.

Взаимодействие MPLAB и Matlab/Simulink позволяет разрабатывать программы для PIC контроллеров в среде Simulink - графического моделирования и анализа динамических систем.

В этой работе рассматриваются средства программирования PIC контроллеров: MPLAB, Matlab/Simulink и программатор PIC-KIT3 в следующих разделах.

- Характеристики миниатюрного PIC контроллера PIC12F629
- Интегрированная среда разработки MPLAB IDE
- Подключение Matlab/Simulink к MPLAB
- Подключение программатора PIC-KIT3

ХАРАКТЕРИСТИКИ МИНИАТЮРНОГО PIC КОНТРОЛЛЕРА

Семейство PIC12xxx содержит контроллеры в миниатюрном 8-выводном корпусе со встроенным тактовым генератором. Контроллеры имеют RISC-архитектуру и обеспечивают выполнение большинства команд процессора за один машинный цикл.

Для примера, ниже даны характеристики недорогого компактного 8-разрядного контроллера PIC12F629 с многофункциональными портами, малым потреблением и широким диапазоном питания [1].

Архитектура	RISC
Напряжение питания V_{DD}	от 2,0В до 5,5В (< 6,5В)
Потребление:	<1,0 мА @ 5,5В, 4МГц 20 мкА (тип) @ 32 кГц, 2,0В <1,0 мкА (тип) в режиме SLEEP@2,0В
Рассеиваемая мощность	0,8Вт
Многофункциональные каналы ввода/вывода	6/5
Максимальный выходной ток портов GPIO	125мА
Ток через программируемые внутренние подтягивающие резисторы портов	≥ 50 (250) ≤ 400 мкА @ 5,0В
Разрядность контроллера	8
Тактовая частота от внешнего генератора	20 МГц
Длительность машинного цикла	200 нс
Тактовая частота от внутреннего RC генератора	4 МГц +/-1%
Длительность машинного цикла	1 мкс
FLASH память программ Число циклов стирание/запись	1кБайт ≥ 1000
ОЗУ память данных	64 байта
EEPROM память данных Число циклов стирание/запись	128 $\geq 10K$ ($-40^{\circ}C \leq T_A \leq +125^{\circ}C$)
Аппаратные регистры специального назначения	16
Список команд	35 инструкций, все команды выполняются за один машинный цикл, кроме команд перехода, выполняемых за 2 цикла
Аппаратный стек	8 уровней
Таймер/счетчик TMR0	8-разрядный с предделителем
Таймер/счетчик TMR1	16-разрядный с предделителем
Предельная рабочая температура для E исполнения (расширенный диапазон)	от $-40^{\circ}C$ до $+125^{\circ}C$
Температура хранения	от $-65^{\circ}C$ до $+150^{\circ}C$

Дополнительные особенности контроллера:

- Сброс по включению питания (POR)
- Таймер сброса (PWRT) и таймер ожидания запуска генератора (OST)
- Сброс по снижению напряжения питания (BOD)
- Сторожевой таймер WDT
- Мультиплексируемый вывод -MCLR
- Система прерываний по изменению уровня сигнала на входах
- Индивидуально программируемые для каждого входа подтягивающие резисторы
- Программируемая защита входа
- Режим пониженного энергопотребления SLEEP
- Выбор режима работы тактового генератора
- Внутрисхемное программирование ICSP с использованием двух выводов
- Четыре пользовательские ID ячейки

КМОП технология контроллера обеспечивает полностью статический режим работы, при котором остановка тактового генератора не приводит к потере логических состояний внутренних узлов.

Микроконтроллер PIC12F629 имеет 6-разрядный порт ввода/вывода GPIO. Один вывод GP3 порта GPIO работает только на вход, остальные выходы можно сконфигурировать для работы как на вход так и на выход. Каждый вывод GPIO имеет индивидуальный бит разрешения прерываний по изменению уровня сигнала на входах и бит включения внутреннего подтягивающего резистора.

ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ MPLAB IDE

MPLAB IDE - бесплатная интегрированная среда разработки ПО для микроконтроллеров PIC включает средства для создания, редактирования, отладки, трансляции и компоновки программ, записи машинного кода в микроконтроллеры через программаторы.

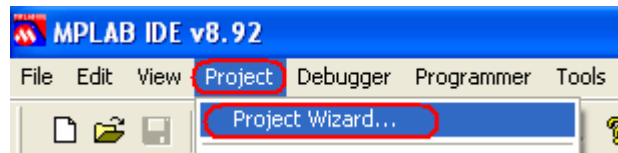
Загрузка MPLAB IDE

Бесплатные версии MPLAB (включая MPLAB 8.92) хранятся на сайте компании Microchip <http://www.microchip.com/mplab/mplab-x-ide> > **DOWNLOAD ARCHIVE**

Создание проекта

Пример создания проекта программ PIC контроллера в среде MPLAB включает следующие шаги [2].

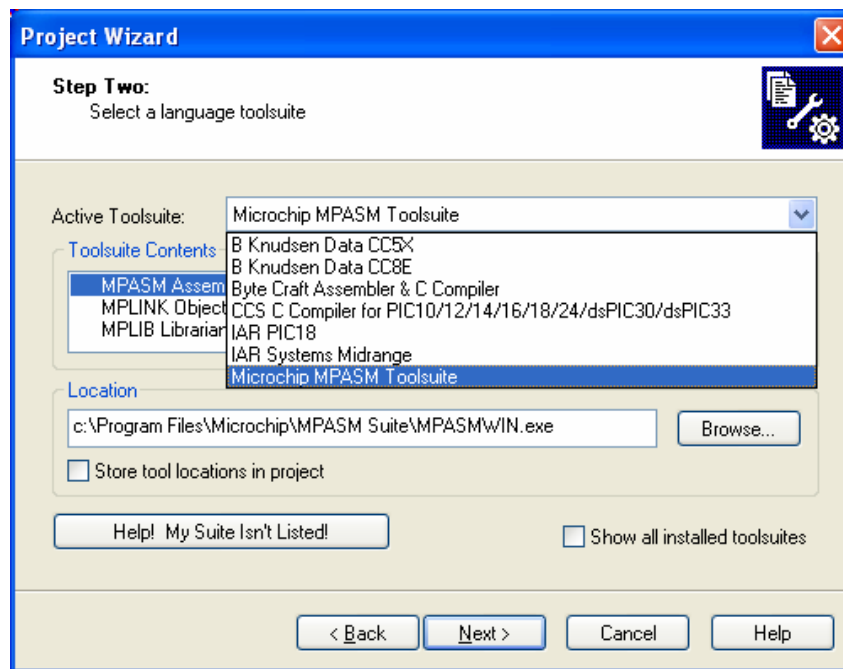
1. Вызов менеджера проекта.



2. Выбор типа PIC микроконтроллера.



3. Выбор компилятора, например, Microchip MPASM для ассемблера.



4. Выбор пути к каталогу проекта (клавиша **Browse...**) и ввод имени проекта.
5. Подключение файлов к проекту в окне Project Wizard > **Step Four** можно не выполнять. Это можно сделать позднее, внутри активного проекта. Клавиша **Next** открывает следующее окно.
6. Завершение создания проекта (клавиша **Finish**).

В результате создания проекта FirstPrMPLAB интерфейс MPLAB принимает вид, показанный на Рис. 1.

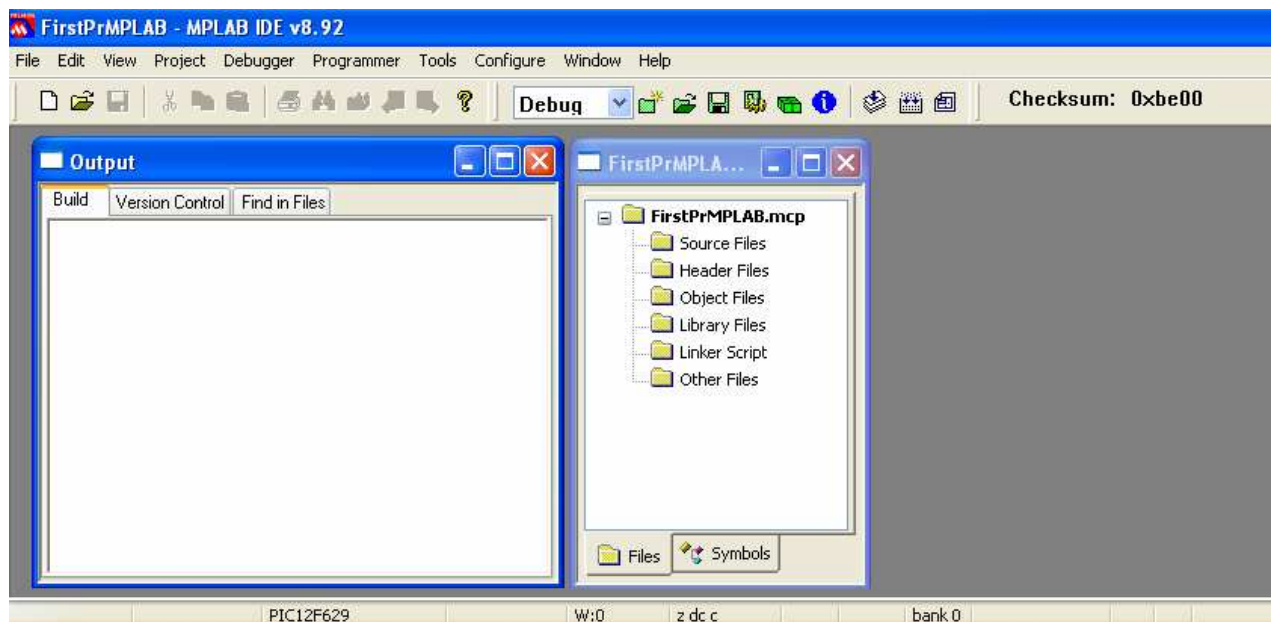


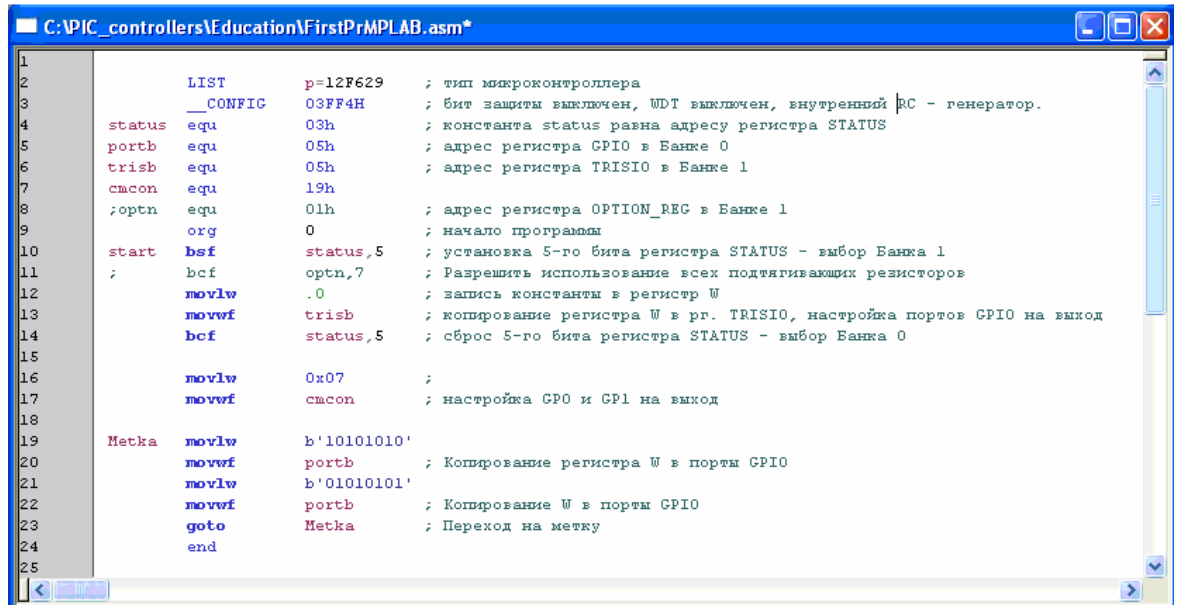
Рис. 1. Интерфейс среды MPLAB v8.92 и шаблон проекта.

Создание файла программы

Программу можно создать при помощи любого текстового редактора. В MPLAB имеется встроенный редактор, который обеспечивает ряд преимуществ, например, оперативный лексический анализ исходного текста, в результате которого в тексте цветом выделяются зарезервированные слова, константы, комментарии, имена, определенные пользователем.

Создание программы в MPLAB можно выполнить в следующей последовательности.

1. Открыть редактор программ: меню > File > **New**. Изначально программе присвоено имя **Untitled**.
2. Набрать или скопировать программу, например, на ассемблере.
3. Сохранить программу под другим именем (меню > File > **Save As**), например, FirstPrMPLAB.asm.



```
1
2          LIST      p=12F629      ; тип микроконтроллера
3          __CONFIG  03FF4H      ; бит защиты выключен, WDT выключен, внутренний RC - генератор.
4          status   equ    03h      ; константа status равна адресу регистра STATUS
5          portb    equ    05h      ; адрес регистра GPIO в Банке 0
6          trisb    equ    05h      ; адрес регистра TRISIO в Банке 1
7          cmcon    equ    19h
8          ;optn    equ    01h      ; адрес регистра OPTION_REG в Банке 1
9          org      0              ; начало программы
10         start    bsf     status,5 ; установка 5-го бита регистра STATUS - выбор Банка 1
11         ;        bcf     optn,7   ; Разрешить использование всех подтягивающих резисторов
12         movlw    .0              ; запись константы в регистр W
13         movwf    trisb          ; копирование регистра W в рег. TRISIO, настройка портов GPIO на выход
14         bcf     status,5        ; сброс 5-го бита регистра STATUS - выбор Банка 0
15
16         movlw    0x07           ;
17         movwf    cmcon          ; настройка GP0 и GP1 на выход
18
19         Metka    movlw    b'10101010'
20                 movwf    portb   ; Копирование регистра W в порты GPIO
21                 movlw    b'01010101'
22                 movwf    portb   ; Копирование W в порты GPIO
23                 goto     Metka   ; Переход на метку
24         end
25
```

Рис. 2. Пример простейшей программы (на ассемблере) вывода сигналов через порты контроллера GP0, GP1, GP2, GP4, GP5 на максимальной частоте.

Запись '1' в разряде регистра TRISIO переводит соответствующий выходной буфер в 3-е состояние, в этом случае порт GP может работать только на вход. Установка нуля в TRISIO настраивает работу порта GP на выход.

Примечание. По спецификации PIC12F629 порт GP3 микроконтроллера работает только на вход (соответствующий бит регистра TRISIO не сбрасывается – всегда находится в '1').

Регистры TRISIO и GPIO находятся в разных банках области памяти. Переключение банков выполняется 5-м битом регистра STATUS.

Любая программа на ассемблере начинается директивой **org** и заканчивается директивой **end**. Переход **goto Metka** обеспечивает циклическое выполнение программы.

В программе (Рис. 2) используются следующие обозначения.

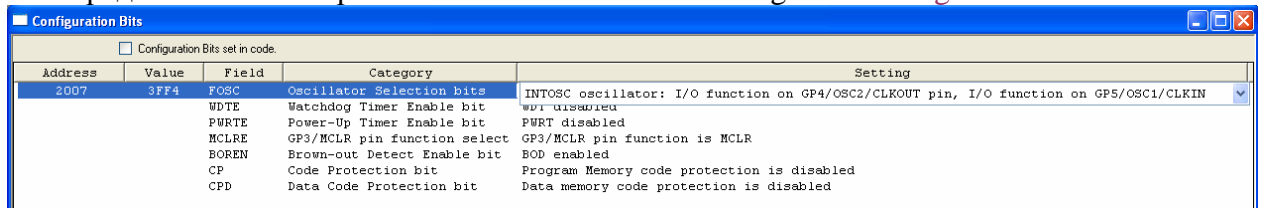
Директива LIST	назначение типа контроллера
Директива __CONFIG	установка значений битов конфигурации контроллера
Директива equ	присвоение числового значения
Директива org 0	начало выполнения программы с адреса 0
Команда bsf	устанавливает бит указанного регистра в 1
Команда bcf	сбрасывает бит указанного регистра в 0
Команда movlw	записывает константу в регистр W
Команда movwf	копирует содержимое регистра W в указанный регистр
Команда goto	обеспечивает переход без условия на строку с меткой
Директива end	конец программы

Установка требуемой конфигурации микроконтроллера

Конфигурация микроконтроллера PIC12F629 зависит от настроек слова конфигурации (2007h), которые можно задать в программе через директиву `__CONFIG`

R/P-1	R/P-1	U - 1	U - 1	U - 1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
BG1	BG0	-	-	-	-CPD	-CP	BODEN	MCLRE	-PWRTE	WDTE	FOSC2	FOSC1	FOSC0
Бит 13												Бит 0	

непосредственно или через окно MPLAB: меню > Configure > Configuration Bits



где

Бит 2-0 FOSC2:FOSC0. Выбор тактового генератора

- 111 Внешний RC генератор. Подключается к выводу GP5. GP4 работает как CLKOUT
- 110 Внешний RC генератор. Подключается к выводу GP5. GP4 работает как ввод/вывод
- 101 Внутренний RC генератор 4МГц. GP5 работает как ввод/вывод. GP4 - как CLKOUT
- 100 Внутренний RC генератор 4МГц. GP5 и GP4 работают как ввод/вывод
- 011 ЕС генератор. GP4 работает как ввод/вывод. GP5 - как CLKIN
- 010 НС генератор. Резонатор подключается к GP4 и GP5
- 001 ХТ генератор. Резонатор подключается к GP4 и GP5
- 000 LP генератор. Резонатор подключается к GP4 и GP5

Бит 3 WDTE: настройка сторожевого таймера (Watchdog Timer)

- 1 WDTE включен
- 0 WDTE выключен

Сторожевой таймер предохраняет микроконтроллер от зависания – перезапускает программу через определенный интервал времени если таймер не был сброшен. Период таймера устанавливается в регистре OPTION_REG. Обнуление сторожевого таймера вызывается командой CLRWDT.

Бит 4 .PWRTE: Разрешение работы таймера включения питания:

- 1 PWRT выключен
- 0 PWRT включен

Таймер задерживает микроконтроллер в состоянии сброса при подаче питания V_{DD} .

Бит 5 MCLR: Выбор режима работы вывода GP3/-MCLR

- 1 работает как -MCLR
- 0 работает как порт ввода-вывода GP3

Бит 6 BODEN: Разрешение сброса по снижению напряжения питания (как правило < 2.0В)

- 1 разрешен сброс BOR
 - 0 запрещен сброс BOR автоматически включается таймер
- При разрешении сброса BOR автоматически включается таймер PWRT

Бит 7 .CP: Бит защиты памяти программ от чтения программатором

- 1 Защита выключена
 - 0 Защита включена
- При выключения защиты вся память программ стирается

Бит 8 .CPD: Бит защиты EPROM памяти данных

- 1 Защита выключена

0 Защита включена
 После выключения защиты вся информация будет стерта

Бит 11-9 Не используются: Читается как '1'

Бит 13-12 BG1:BG0. Биты калибровки сброса по снижению питания

00 нижний предел калибровки

11 верхний предел калибровки

Добавление программы к проекту

Пример добавления программы к проекту показан на (Рис. 3).

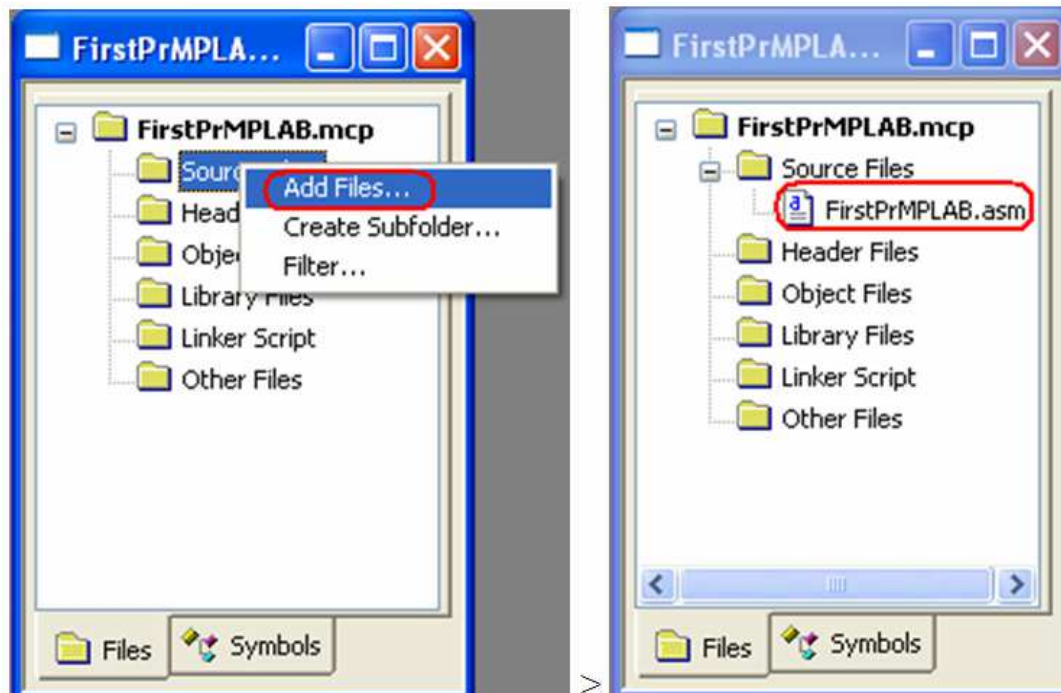


Рис. 3. Добавление программы FirstPrMPLAB.asm к проекту FirstPrMPLAB.mcp

Сохранить материалы проекта можно командой: меню > File > **Save Workspace**.

Компиляция

Чтобы создать бинарный файл с расширением hex для прошивки микроконтроллера необходимо откомпилировать проект. Запуск компиляции выполняется командой меню > Project > **Build All**. Результаты компиляции можно увидеть в окне **Output** (Рис. 1). Если в программе нет ошибок, то компилятор выдаёт сообщение об успешной компиляции: **BUILD SUCCEEDED**, загрузочный HEX файл можно найти в рабочем каталоге:

FirstPrMPLAB	cof	781
FirstPrMPLAB	map	1,815
FirstPrMPLAB	HEX	109
FirstPrMPLAB	o	741
FirstPrMPLAB	err	0
FirstPrMPLAB	lst	3,245
FirstPrMPLAB	mcp	1,005
FirstPrMPLAB	asm	932
FirstPrMPLAB	mcw	32,768

Отладка программы

Отладку программы в среде MPLAB IDE можно выполнить при помощи аппаратного эмулятора MPLAB REAL ICE или программного симулятора MPLAB SIM. Запуск последнего выполняется как показано на Рис. 4.

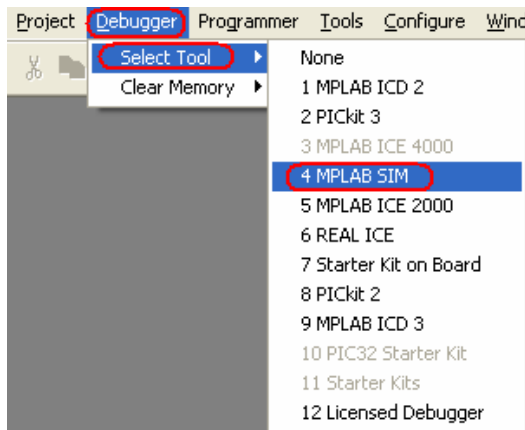


Рис. 4. Подключение к симулятору MPLAB SIM для отладки программы.

После запуска отладчика в окне Output (Рис. 1) появляется закладка MPLAB SIM, куда MPLAB выводит текущую информацию отладчика. Команды отладчика (Рис. 5) после запуска становятся активными.

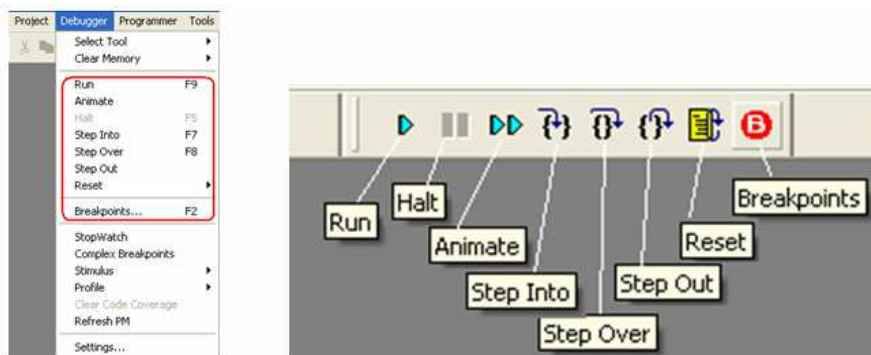




Рис. 5. Команды отладчика.

Команды отладчика:

- Run Непрерывное выполнение программы до точки останова (Breakpoint) если таковая установлена.
- Halt Остановка программы на текущем шаге выполнения.
- Animate Анимация непрерывного выполнения программы.
- Step Into Выполнение по шагам (вызовы Call выполняются за один шаг).
- Step Over Выполнение по шагам включая команды вызовов Call.
- Reset Начальная установка программы. Переход указателя на первую команду.
- Breakpoints Отображение списка точек останова. Обработка списка.

При выполнении программы по шагам текущий шаг выделяется стрелкой  (Рис. 6). Непрерывное выполнение программы останавливается командой Halt или достижением программой точки останова. Точка останова  устанавливается/снимается в строке программы двойным щелчком.

Пример программы на ассемблере, которая с максимальной скоростью меняет состояние портов контроллера показан на Рис. 6 (справа). Программа передаёт в

регистр портов GPIO данные b'10101010' и b'01010101'. Поскольку в регистре GPIO передачу данных в порты контроллера выполняют не все разряды, а только 0,1,2,4 и 5, то состояние регистра GPIO (Рис. 6, слева) отличается значениями: b'00100010' и b'00010101'.

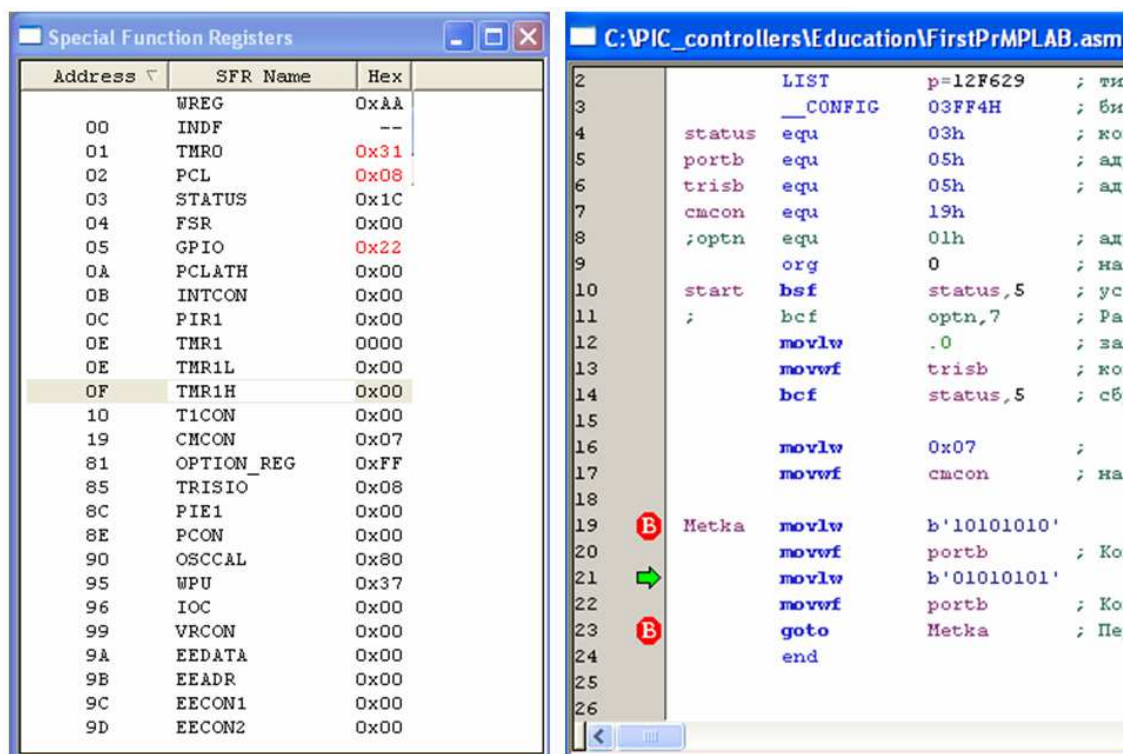


Рис. 6. Состояние регистров специального назначения контроллера на момент выполнения программы (слева) и выполняемая по шагам программа (справа).

В процессе отладки можно наблюдать за состоянием регистров, переменных, памяти в соответствующих окнах, открываемых в разделе View основного меню. В процессе отладки можно вносить изменения в код программы, содержимое регистров, памяти, изменять значения переменных. После изменения кода необходимо перекомпилировать программу. Изменение содержимого регистров, памяти и значения переменных (окна раздела View: Special Function Register, File Register, EEPROM, Watch) не требует перекомпиляции.

Входные сигналы портов модели микроконтроллера можно задать в разделе **Debugger > Stimulus**. Устанавливаемые состояния сигналов портов привязываются к времени (тактам) отладки.

Иногда результаты выполнения программы в режиме отладки не соответствуют выполнению этой же программы в реальном контроллере, так, например, отладчик программы (Рис. 6) без инструкций **movlw 0x07** и **movwf cmcon** показывает, что выходы GP0 и GP1 регистра GPIO не изменяются - находятся в нулевом состоянии, содержимое регистра GPIO попеременно равно 0x14 и 0x20. Однако, контроллер, выполняющий программу без указанных инструкций, показывает на осциллографе циклическую работу всех пяти выходов: 0x15 и 0x22, включая GP0 и GP1 (см. Рис. 7).

Осциллограммы контроллера, выполняющего циклы программы Рис. 6 (Metka ... goto Metka) показаны на Рис. 7.



Рис. 7. Осциллограммы выхода GP0 (слева) и GP1 (справа) микроконтроллера PIC12F629, работающего от внутреннего 4МГц RC генератора. Программа (Рис. 6) формирует сигналы максимальной частоты на всех выходах контроллера. За период сигналов 5.3 мкс выполняется 5 команд (6 машинных циклов), амплитуда GP0 сигнала на осциллограмме равна 4.6В, измеренное программатором питание контроллера 4.75В.

Прошивка микроконтроллера

Для записи программы в микроконтроллер (прошивки контроллера) необходимо микроконтроллер подключить к интегрированной среде MPLAB IDE через программатор. Организация подключения показана ниже в разделе "Подключение программатора PIC-KIT3".

Примечание. В контроллер PIC12F629 записана заводская калибровочная константа настройки частоты внутреннего тактового генератора. При необходимости её можно прочитать и восстановить средствами MPLAB с использованием программатора.

Команды для работы с программатором и изменения его настроек находятся в меню MPLAB **Programmer**. Тип программатора в MPLAB выбирается в разделе: меню > Programmer > **Select Programmer** (Рис. 8).

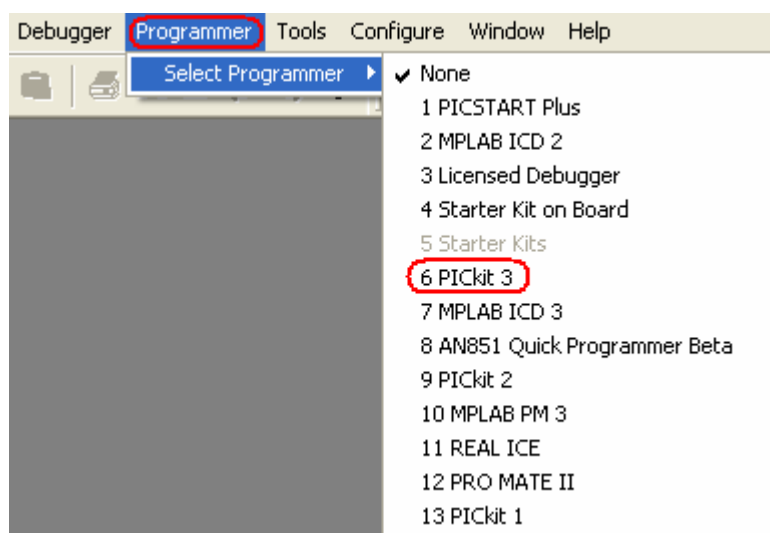


Рис. 8. Выбор программатора для подключения к среде MPLAB.

Прошивка микроконтроллера через программатор запускается командой: меню > Programmer > **Program**. Сообщение об успешной прошивке показано на Рис. 9.

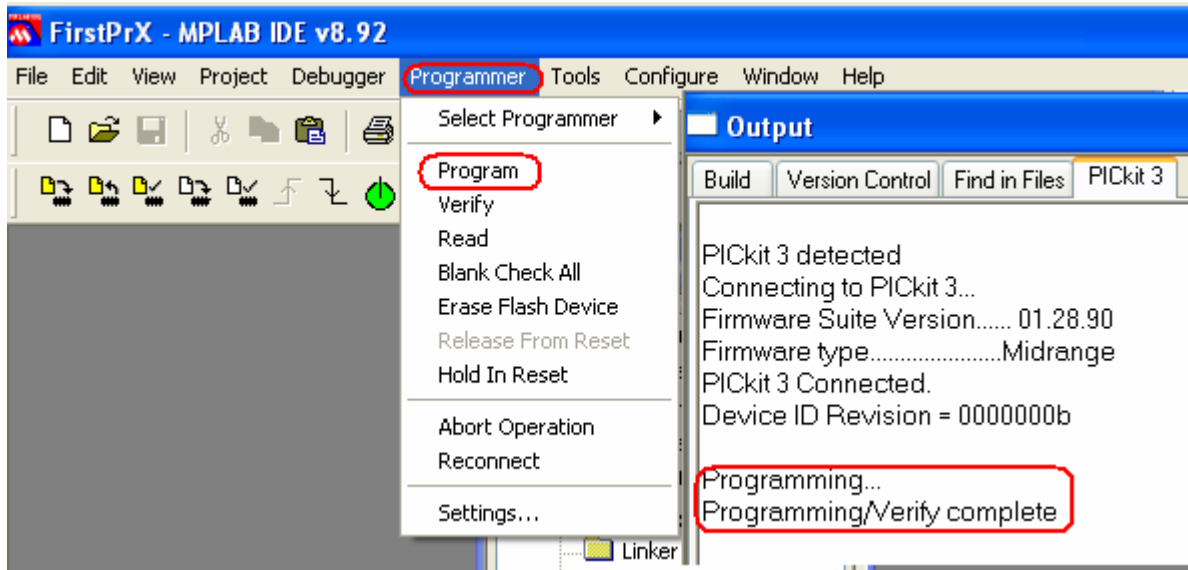


Рис. 9. Запуск прошивки микроконтроллера и вид сообщения об успешной прошивке.

Примечание: Во время прошивки микроконтроллера у программатора PIC-KIT3 мигает желтый светодиод.

ПОДКЛЮЧЕНИЕ MATLAB/SIMULINK К MPLAB

В системе моделирования динамических систем Simulink (приложение к Matlab) на языке графического программирования [8] можно разрабатывать программы для семейства PIC контроллеров имеющих АЦП/ЦАП, счетчики, таймеры, ШИМ, DMA, интерфейсы UART, SPI, CAN, I2C и др.

Пример Simulink программы PIC контроллера показан на Рис. 10.

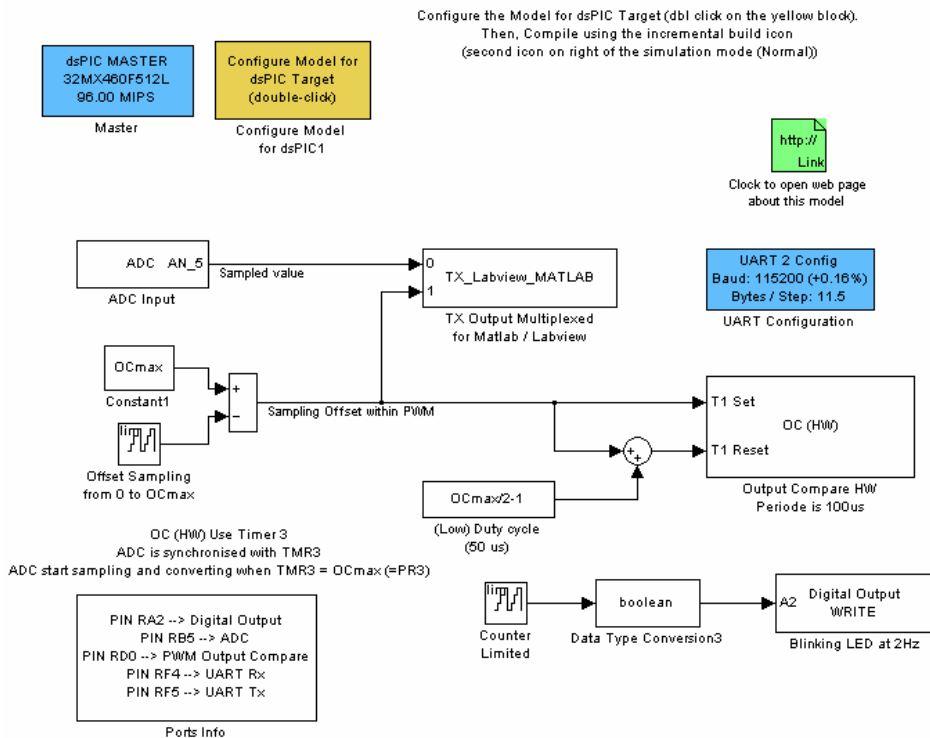


Рис. 10. Пример программы на языке графического программирования для PIC контроллера выполненной в среде моделирования динамических систем Simulink.

Взаимодействие средств разработки и компиляции программ для PIC контроллеров в Simulink показано на Рис. 11 [6].



Рис. 11. Структура средств построения адекватной модели PIC контроллера на языке графического программирования.

Для построения среды разработки необходимы следующие компоненты Matlab

- Simulink
- Real-Time Workshop Embedded Coder
- Real-Time Workshop

и Си компилятор компании Microchip:

- C30 для контроллеров PIC24, dsPIC30 и PIC33
- или C32 для контроллеров серии PIC32

Установка компонентов Matlab


На сайте <http://www.kerhuel.eu/wiki/Download#Limitation> имеются Simulink библиотеки (dsPIC Toolbox) для PIC контроллеров и версий Matlab с R2006a по R2012a:

- [Download dsPIC Toolbox for Matlab R2012a](#) (Matlab 32 and 64 bits)
- • •
- [Download dsPIC Toolbox for Matlab R2006a](#) (Matlab 32 bits) (31/03/2010)

Для скачивания библиотеки необходимо зарегистрироваться. Программы поддерживают работу 100 микроконтроллеров из серий PIC 16MC, 24F, 30F, 32MC, 33F, 56GP, 64MC, 128MC, 128GP.

Бесплатные версии работают с Simulink моделями PIC контроллеров имеющих до 7 портов ввода-вывода.

Для установки dsPIC Toolbox - библиотеки блоков PIC контроллеров для Matlab/Simulink необходимо [4]:

- Скачать dsPIC Toolbox для требуемой версии Matlab.
- Распаковать zip файл в папке, в которой будут установлены Simulink блоки.
- Запустить Matlab.
- Настроить текущий каталог Matlab на папку с распакованным файлом.
- Открыть и запустить файл `install_dsPIC_R2012a.m`, например, кнопкой  меню или клавишей <F5> клавиатуры.

Библиотеки dsPIC и примеры Simulink моделей устанавливаются в текущую папку Matlab (Рис. 12). Установленные блоки для моделирования PIC контроллеров доступны в разделе **Embedded Target for Microchip dsPIC** библиотеки Simulink (Рис. 13).

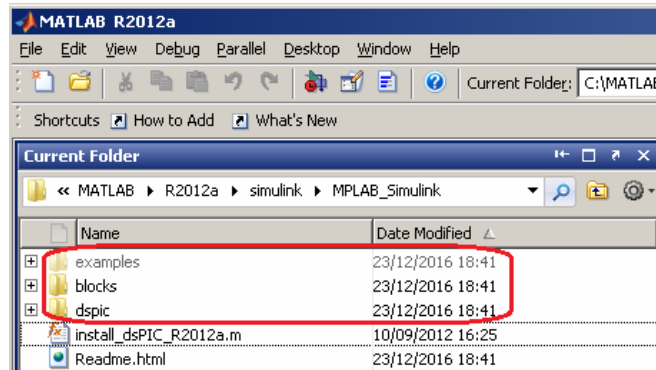


Рис. 12. Содержимое текущего каталога после выполнения `install_dsPIC_R2012a.m`.

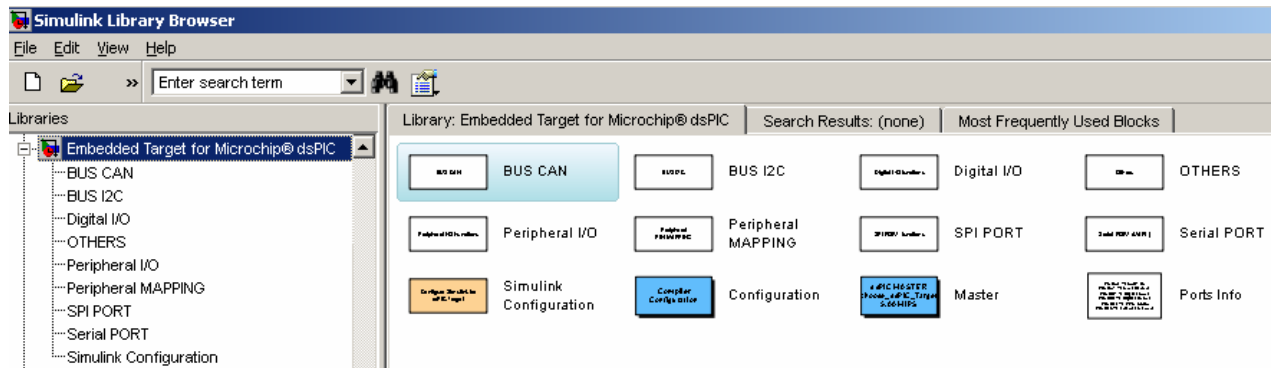


Рис. 13. Блоки, установленной библиотеки "Embedded Target for Microchip dsPIC".

Для совместной компиляции Simulink модели средствами Matlab и MPLAB необходимо прописать в переменной окружения `path` Matlab с высшим приоритетом путь к каталогу MPLAB с файлами `MplabOpenModel.m`, `MplabGetBuildinfo.m` и `getHardwareConfigs.m`:

```
>>path('c:\Program Files (x86)\Microchip\MPLAB IDE\Tools\MATLAB',path)
```

Установка Си компилятора MPLAB

Компиляторы MPLAB находятся на сайте Microchip:

<http://www.microchip.com/mplab/mplab-x-ide> > Download Archive > MPLAB C Compiler for PIC24 and dsPIC DSCs.

Для установки демонстрационной версии компилятора C30 необходимо его скачать по ссылке **PIC24/dsPIC v3.25** (Рис. 14) и запустить принятый файл `mplabc30-v3.25-comboUpgrade.exe`.



Рис. 14. Версии Си компилятора (слева) и режимы его установки (справа).

Примечание. Работа выполнена с версией **v3.25** компилятора C30 для PIC24/dsPIC. Проверка показала, что следующая версия **v3.30** не поддерживает совместную компиляцию моделей Matlab R2012a (dsPIC Toolbox) без ошибок.

Установочный **exe** файл создаёт в разделе **c:\Program Files (x86)\Microchip** новый каталог **mplabc30** с файлами (Рис. 15).

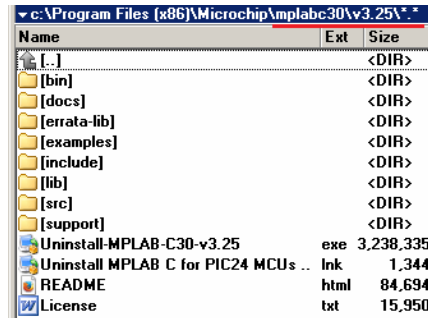


Рис. 15. Каталоги компилятора C30 MPLAB.

Последовательность Simulink программирования для PIC контроллеров

1. Создайте рабочий каталог и скопируйте в него ***.mdl** примеры из раздела **example** (см. Рис. 12).
2. Загрузите Matlab. Настройте его на рабочий каталог.
3. Включите в переменную окружения path Matlab с высшим приоритетом путь к MPLAB - каталогу **c:\Program Files (x86)\Microchip\MPLAB IDE\Tools\MATLAB**:

```
>>path('c:\Program Files (x86)\Microchip\MPLAB IDE\Tools\MATLAB\',path)
```

Примечание: Использование команды **>>path** без аргументов приводит к отображению списка путей переменной path в окне команд (Command Window). Удалить путь из переменной path можно командой **rmpath**, например,

```
>>rmpath(' c:\Program Files\Microchip\MPLAB IDE\Tools\MATLAB\')
```

4. Создайте Simulink модель для PIC контроллера, используя блоки библиотеки "Embedded Target for Microchip dsPIC" (Рис. 13), или загрузите готовую модель, например, Servo_ADC.mdl.

Тип контроллера, для которого разрабатывается Simulink модель, выбирается из списка в блоке **Master** > PIC (Рис. 16, Рис. 10), который должен быть включен в состав модели.

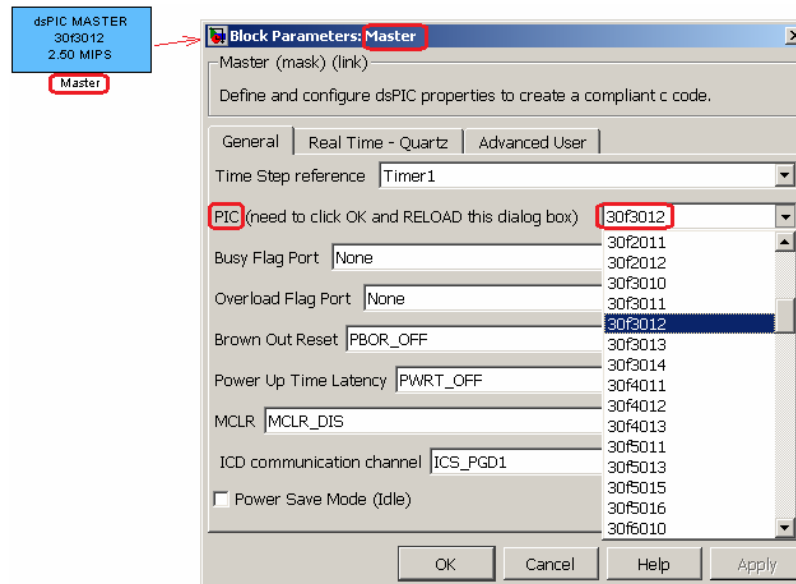


Рис. 16. Выбор типа контроллера в блоке Master модели.

5. Проверьте настройки конфигурации модели: Меню > Simulation > Configuration Parameters <Ctrl+E>. В строке ввода System target file раздела Code Generation должен быть указан компилятор S-функций dspic.tlc (Рис. 17). Выбор dspic.tlc настраивает все остальные параметры конфигурации модели, включая шаг и метод интегрирования.

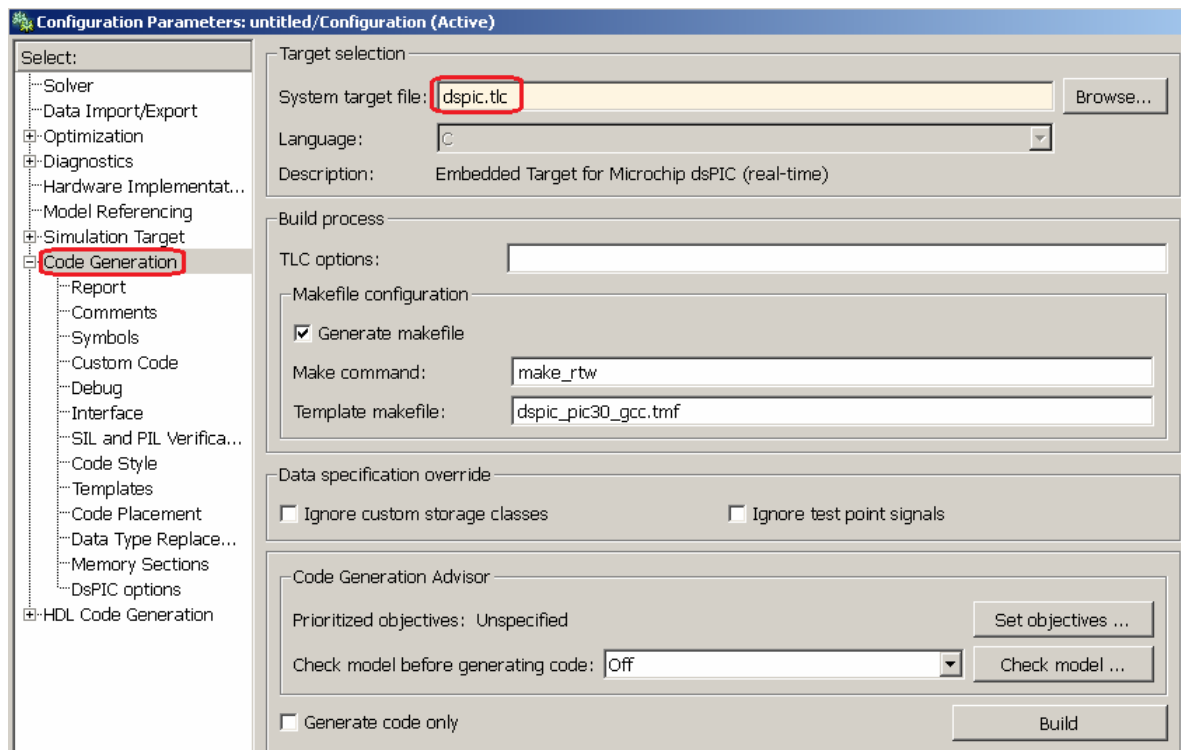


Рис. 17. Выбор компилятора S-функций dspic.tlc для моделей PIC контроллеров в разделе "основное меню > Simulation > Configuration Parameters > Code Generation".

6. Откомпилируйте модель tmp_Servo_ADC.mdl. Запуск компилятора показан на Рис. 18.

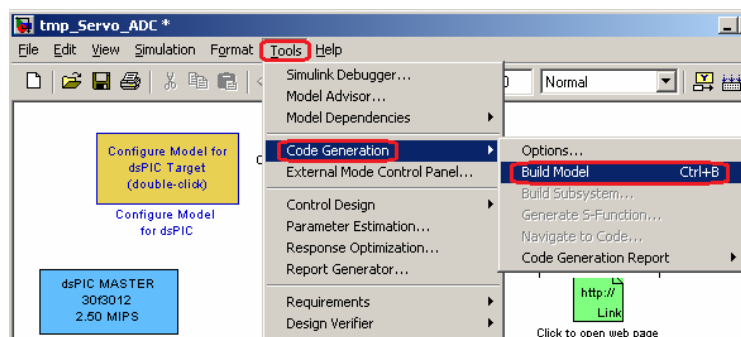


Рис. 18. Запуск компилятора Simulink модели.

В результате успешной компиляции (сообщение: ### Successful completion of build procedure for model: Servo_ADC) в текущем каталоге создаются **HEX** файл для прошивки PIC контроллера и **MCP** проект среды MPLAB (Рис. 19).

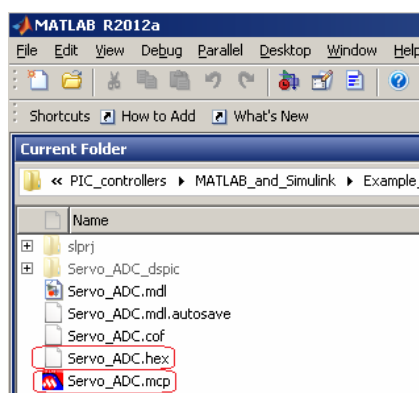




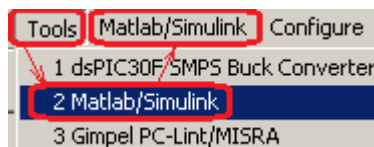
Рис. 19. Результаты компиляции модели.

Запуск модели в Matlab/Simulink выполняется в окне модели кнопкой , условное время моделирования устанавливается в строке .

Управление компиляцией Simulink моделей из среды MPLAB

Управление компиляцией Simulink модели можно выполнять командами раздела **Matlab/Simulink** среды MPLAB, например, в следующем порядке.

1. Разработайте модель PIC контроллера в Matlab/Simulink. Сохраните модель.
2. Запустите **MPLAB**.
3. Выберите MPLAB меню > Tools > **Matlab/Simulink** и новый раздел появится в составе меню.



4. В разделе **Matlab/Simulink** откройте Simulink модель, например, Servo_ADC, командой "Matlab/Simulink > Specify Simulink Model Name > Open > File name > Servo_ADC.mdl > **Open**". Команда Open запускает Matlab и открывает модель.
5. Откомпилируйте модель и создайте **MCP** проект командами **Generate Codes** или **Generate Codes and Import Files**. Перевод MDL модели в MCP проект выполняется TLC компилятором Matlab.

В результате создаётся проект MPLAB  **Servo_ADC.mcp** со скриптами модели на языке Си.

6. Откройте проект: меню > Project > Open > **Servo_ADC.mcp** (Рис. 20).

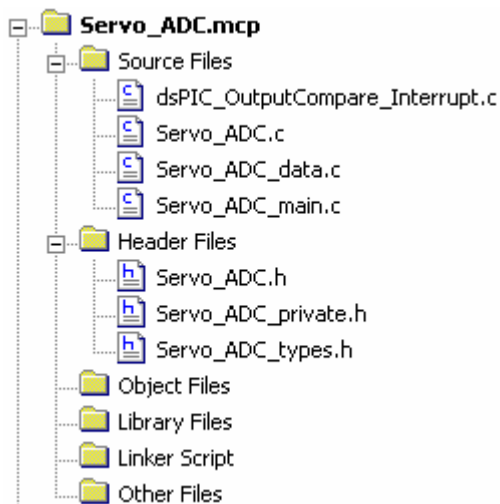


Рис. 20. Структура MCP проекта Simulink модели Servo_ADC.mdl в среде MPLAB.

Проект Simulink модели готов для редактирования, отладки и компиляции в машинные коды контроллера средствами MPLAB.

ПОДКЛЮЧЕНИЕ ПРОГРАММАТОРА PIC-KIT3

Узнать какие программаторы записывают бинарный код в конкретный микроконтроллер можно в разделе меню > Configure > **Select Device** среды MPLAB 8.92. Например, программатор PIC-KIT3 не поддерживает контроллер PIC12C508A (Рис. 21, левый рисунок), но работает с контроллером PIC12F629 (Рис. 21, правый рисунок).

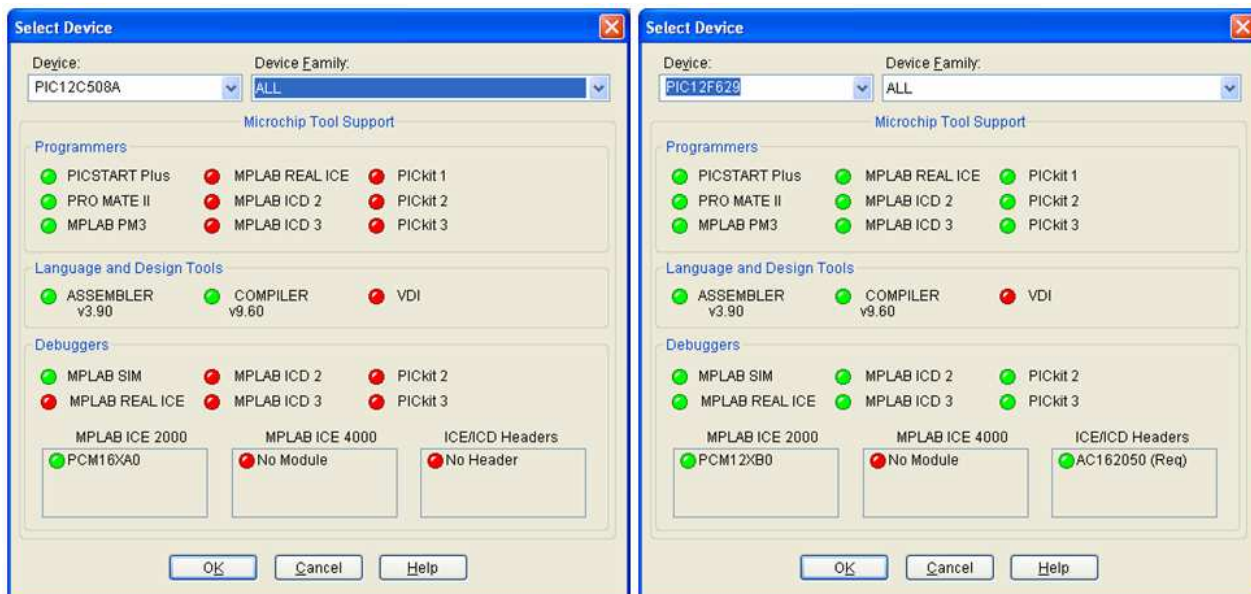


Рис. 21. Перечень программаторов для прошивки микроконтроллера.

Информацию об установленном драйвере программатора PIC-KIT3 можно запросить у менеджера устройств ОС Windows (Рис. 22).

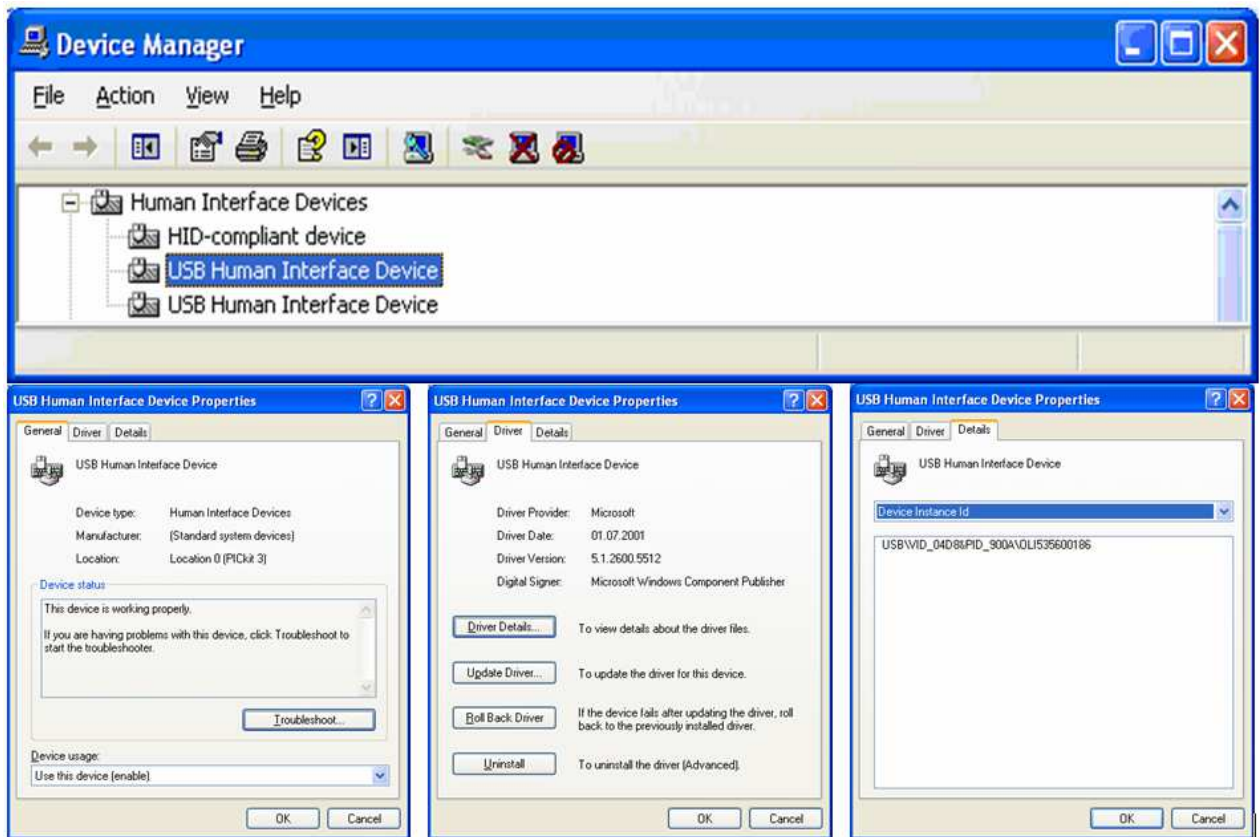


Рис. 22. Информация об установленном драйвере программатора PIC-KIT3.

Схема подключения микроконтроллера PIC12F629 к программатору PIC-KIT3 показана на Рис. 23.

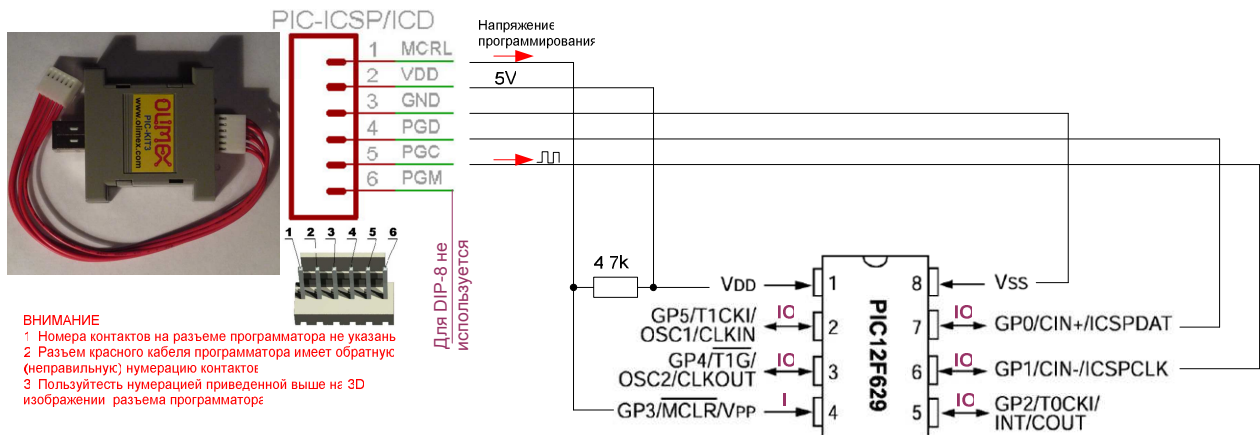


Рис. 23. Схема подключения микроконтроллера PIC12F629 к программатору PIC-KIT3.

Вывод PGM программатора для прошивки контроллеров PIC12F629 не используется. Наличие вывода PGM для разных типов PIC контроллеров показано на Рис. 24. Вывод PGM рекомендуется "притягивать" к общему проводу (GND), через резистор, номиналом 1K [3].

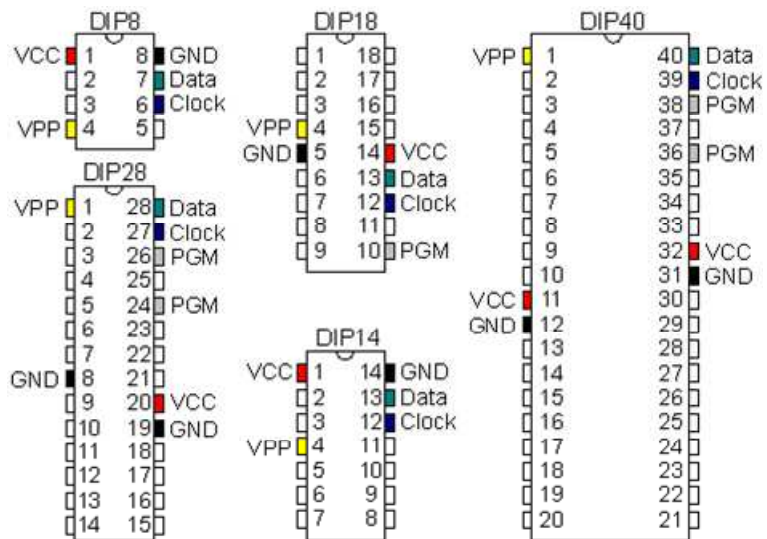


Рис. 24. Выводы PGM PIC контроллеров.

Индикация светодиодов программатора Olimex PIC-KIT3 показана в следующей таблице.

Желтый	Красный	Состояние программатора
Вкл	Выкл	Подключен к USB линии
Вкл	Вкл	Взаимодействие с MPLAB
Мигает	Включен постоянно	Прошивка микроконтроллера

Не следует подключать питание микроконтроллера V_{DD} (Рис. 23) к программатору, если контроллер запитывается от своего источника питания.

При питании микроконтроллера от программатора на линии V_{DD} необходимо установить рабочее напряжение, например, 5В программой MPLAB (Menu> Programmer> Settings > Power), как показано на Рис. 25.

Примечание. При отсутствии напряжения на линии V_{DD} MPLAB IDE выдает сообщение об ошибке: **PK3Err0045: You must connect to a target device to use**

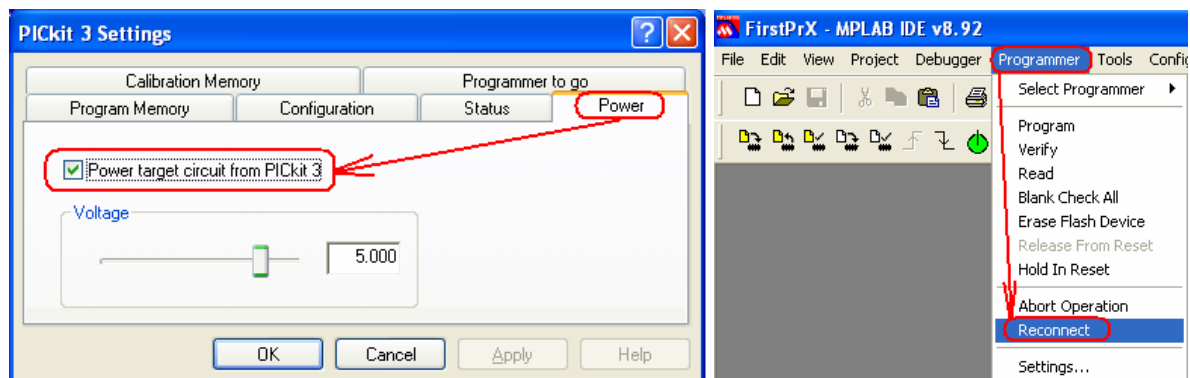


Рис. 25. Установка напряжения V_{DD} на программаторе PIC-KIT3 программой MPLAB IDE v8.92.

Если программатор не может установить требуемое напряжение, например, 5В при его питании от USB, в которой напряжение меньше 5В, MPLAB IDE выдает сообщение об ошибке: **PK3Err0035: Failed to get Device ID**. В этом случае, сначала необходимо измерить напряжение программатора - считать его в закладке меню> Programmer> Settings > Status, а

затем установить напряжение (не больше измеренного) в закладке меню > Programmer > Settings > Power (Рис. 26).

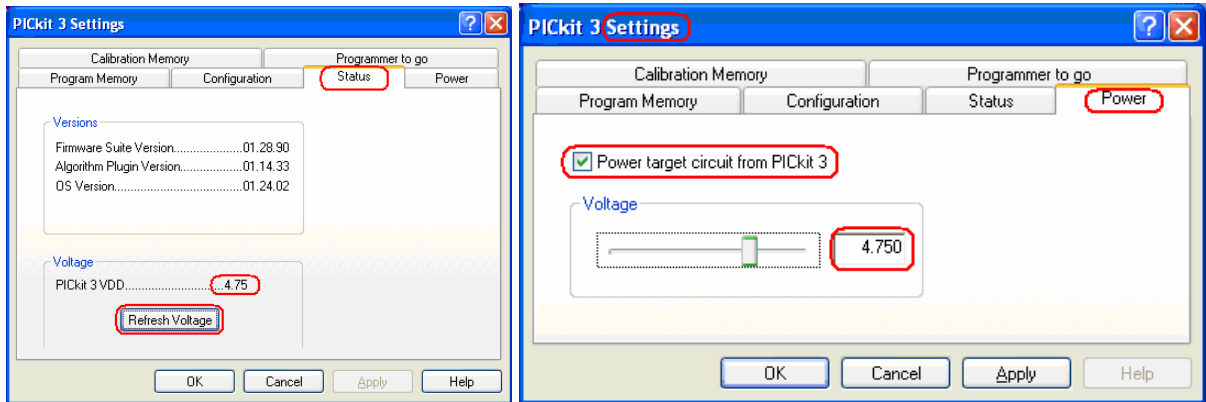


Рис. 26. Измерение (слева) и установка (справа) V_{DD} напряжения программатора PIC-KIT3 программой MPLAB IDE v8.92.

Пример MPLAB сообщения успешного подключения микроконтроллера к программатору по команде меню > Programmer > Reconnect показан на Рис. 27.

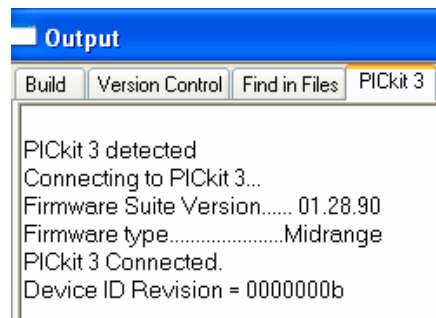


Рис. 27. Сообщение MPLAB об успешном подключении микроконтроллера к программатору.

Можно программировать не только отдельный PIC контроллер, но и контроллер, находящийся в составе рабочего устройства. Для программирования PIC контроллера в составе устройства необходимо предусмотреть установку перемычек и токоограничивающих резисторов как показано на Рис. 28 [3].

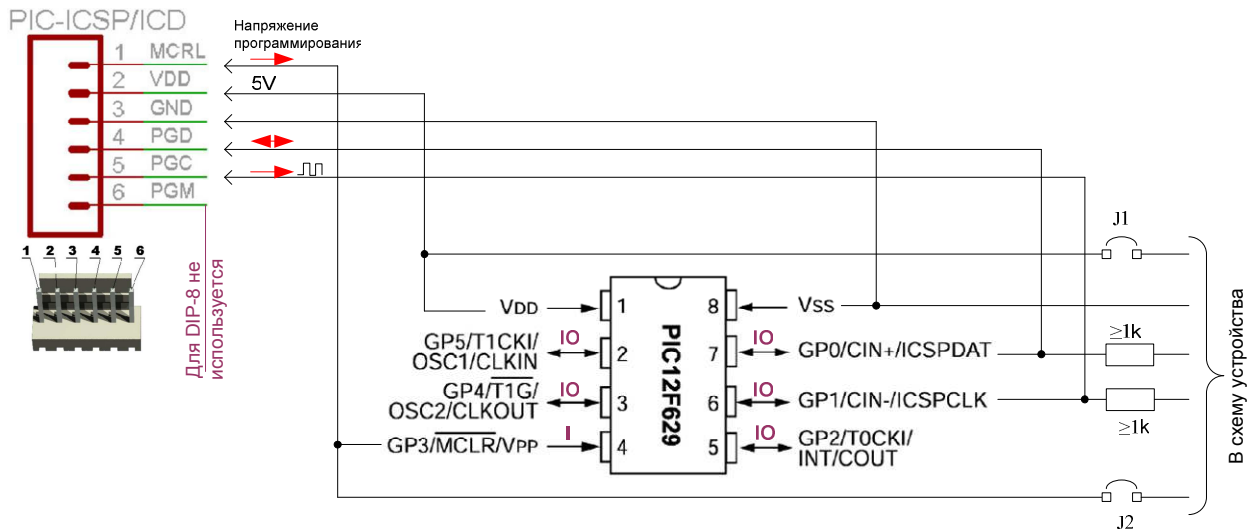


Рис. 28. Подключение микроконтроллера в составе электронного устройства к программатору.

ЗАКЛЮЧЕНИЕ

Малоразрядные PIC-контроллеры имеют широкий диапазон питания, низкое потребление и малые габариты. Они программируются на языках низкого уровня. Разработка программ на языке графического программирования Simulink с использованием многочисленных библиотек значительно сокращает время разработки и отладки в сравнении с программированием на уровне ассемблера. Разработанные для PIC-контроллеров Simulink структуры можно использовать и для компьютерного моделирования динамических систем с участием контроллеров. Однако, из-за избыточности кода такой подход применим только для семейств PIC контроллеров с достаточными ресурсами.

ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Задание 1. Разработка программы генератора импульсов для контроллера PIC12F629.

1. Установите интегрированную среду для программирования PIC контроллеров: MPLAB IDE 8.92 (см. раздел [Загрузка MPLAB IDE](#)).
2. Создайте проект (см. раздел [Создание проекта](#)).
3. Создайте ассемблерную программу генератора импульсов, например, показанную в разделе [Создание файла программы](#), Рис. 2. Используя руководство пользователя [7] рассмотрите назначение используемых в программе команд.
4. Включите программу в проект (Рис. 3).
5. Откомпилируйте программу (см. раздел [Компиляция](#)). По сообщению **BUILD SUCCEEDED** в окне **Output > Build** убедитесь, что компиляция прошла успешно, при необходимости исправьте ошибки.
6. Запустите отладчик **MPLAB SIM** (Рис. 4).
7. Откройте окно отображения состояния специальных регистров процессора (Рис. 6).
8. Откройте окно отображения состояния переменных контроллера **View > Simulator Logic Analyzer**. Включите в окно переменные состояния портов контроллера **GP0 .. GP5** (Рис. 29).

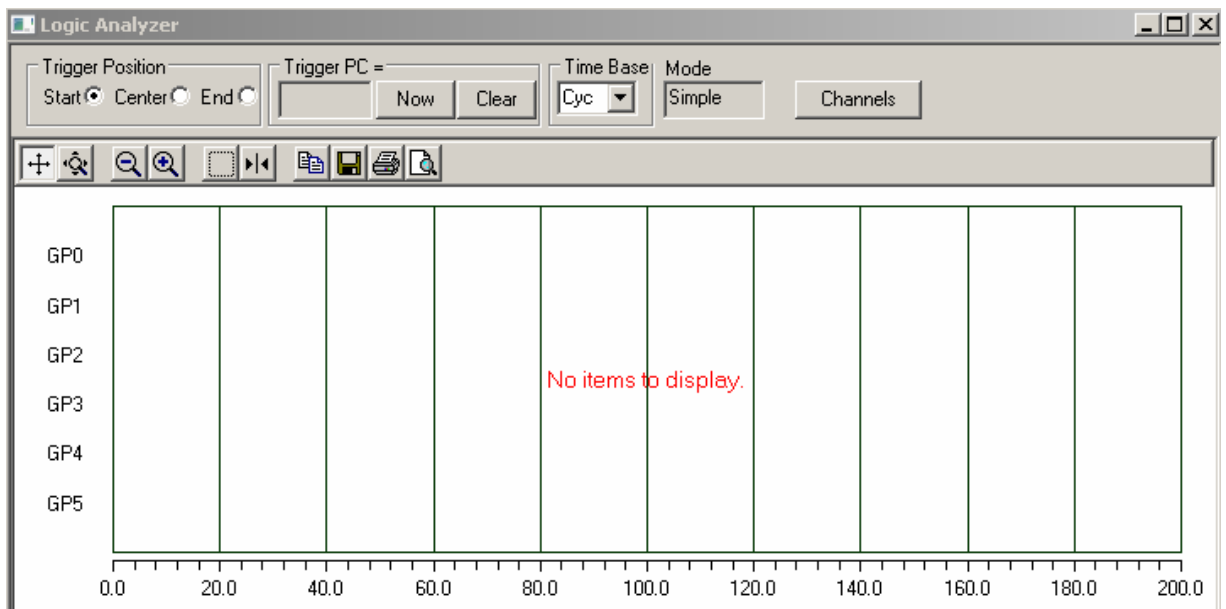


Рис. 29. Окно "осциллографа" Logic Analyzer.

9. Откройте окно отображения состояния переменных программы (View > Watch) и выберите из списка переменную (Рис. 30).

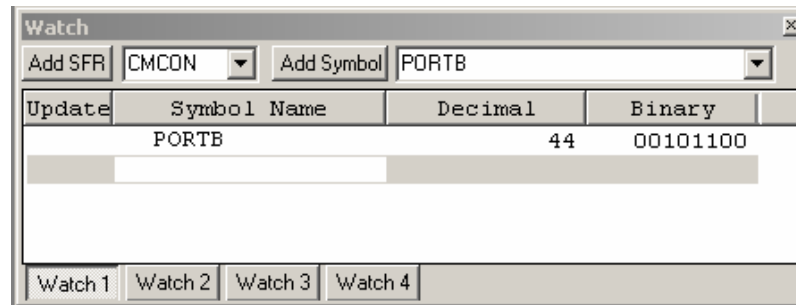


Рис. 30. Окно отображения состояния переменных.

10. Выполняя программу по шагам (команда **Step Into**, Рис. 5) наблюдайте за работой виртуального контроллера по состоянию его регистров и портов (Рис. 6, Рис. 30, Рис. 31).
11. Измерьте период выходного сигнала в машинных циклах. Машинный цикл занимает один такт осциллограммы (Рис. 31).

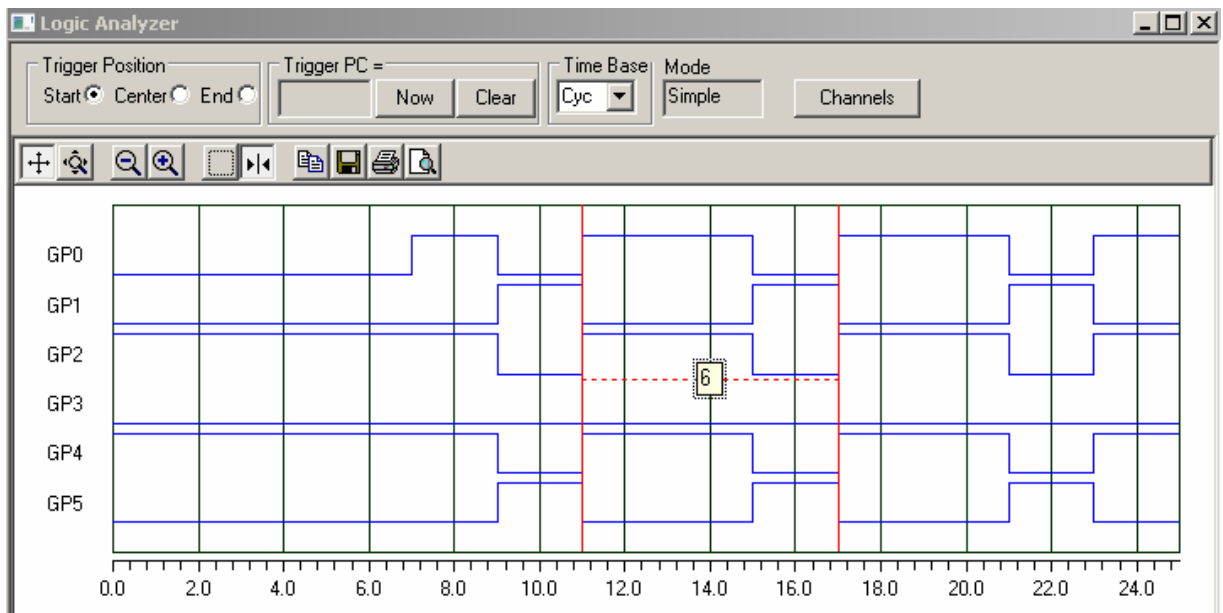


Рис. 31. Временные диаграммы внешних сигналов контроллера. Измерение временных интервалов.

12. Замените блок кода

```

Metka          movlw    0x55;
               movwf   portb;
               movlw   b'10101010';
               movwf   portb;
               movlw   b'01010101';
               movwf   portb;
               goto    Metka
на             movlw    0xFF;
               Metka  |
               xorwf   portb, 1;
               goto    Metka

```

13. Откомпилируйте и выполните по шагам программу. Обратите внимание на изменение формы сигнала.

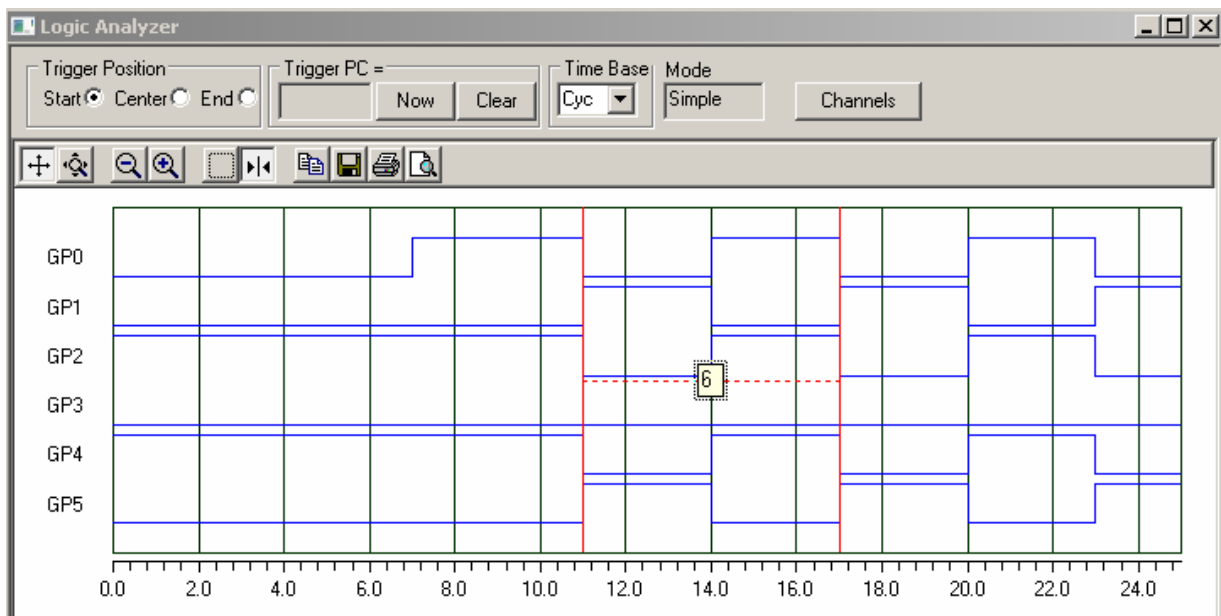
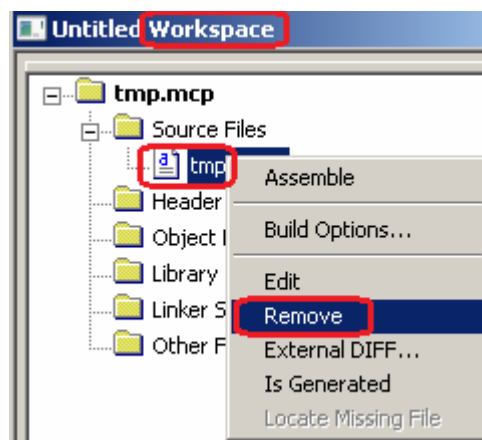


Рис. 32. Меандр на выходе GP0, GP1, GP2, GP5.

14. Найдите и определите размер одноименного бинарного HEX файла, загружаемого через программатор в контроллер.
15. Сохраните текущее состояние интерфейса проекта: **File > Save Workspace**.

Задание 2. Обработка прерываний.

1. Исключите из проекта программу предыдущего задания.



2. Откройте окно редактора новой программы.



3. Создайте программу включающую блок обработки прерываний по входу GP3, например, как показано ниже.

```

LIST      p=12F629
_CONFIG  03E5CH
; 0000 0011 UUUU 0101 0100
; Тип микроконтроллера
; Слово конфигурации (2007h)
; (Бит 13:12) == 11. Установлен верхний предел калибровки по снижению напряжения питания
; (Бит 11:9) Не используется: Читается как '1'
; (Бит 8) = 0. Защита EEPROM памяти данных включена
; (Бит 7) = 0. Защиты памяти программ включена
; (Бит 6) = 1. Разрешен сброс по снижению напряжения питания
; (Бит 5) = 0. GP3/-MCLR работает как цифровой канал порта ввода/вывода

```

```

; (Бит 4) = 1. Таймер включения питания PWRT выключен, +72 мсек на сброс
; (Бит 3) = 0. Сторожевой таймер WDT выключен
; (Бит 2:0) = 100. Работает внутренний RC - генератор

; Регистр STATUS
STATUS EQU 03H ; Адрес регистра STATUS, банк 0, банк 1
Z EQU 2 ; Флаг (регистр STATUS) нулевого результата
DC EQU 1 ; Флаг (регистр STATUS) переноса из младшего полубайта
C EQU 0 ; Флаг (регистр STATUS) переноса из старшего бита
RPO EQU 5 ; Бит переключения банков памяти, 5-й бит регистра STATUS, название RPO

W EQU 0 ; Бит направления результата в аккумулятор
F EQU 1 ; Бит направления результата в регистр

;-----
; Управление портом
PORTB EQU 05H ; Адрес регистра GPIO, банк 0
TRISB EQU 05H ; Адрес регистра TRISIO, банк 1

CMCON EQU 19H ; Адрес регистра CMCON - управление модулем компараторов, банк 0
; без выключения компараторов (CM0:CM2) сигнал единицы на GP0
; держится всего один цикл

GP3 EQU 3 ; Порт GP3, input

;-----
; Переменные обработчика прерываний
W_TEMP EQU 40H ; Регистр сохранения содержимого W при прерываниях
STATUS_TEMP EQU 41H ; Регистр сохранения содержимого STATUS при прерываниях

INTCON EQU 0BH ; Регистр битов разрешения и флагов прерываний
GIE EQU 7 ; Бит глобального разрешения прерываний
GPIE EQU 3 ; Бит разрешения прерываний по изменению сигнала на входах GPIO

IOCB EQU 16H ; Регистр прерываний по изменению уровня сигнала на входах PIC контроллера, Банк 1
IOCB3 EQU 3 ; Бит разрешения прерываний по изменению уровня сигнала на входе GP3

;-----
; Начало программы
;-----
ORG 0 ; Начать выполнение программы
GOTO START ; с метки START

;-----
; Точка входа в прерывание.
;-----
ORG 4 ; Назначение вектора прерывания, необходимо когда используются прерывания

;-----
; НАЧАЛО ОБРАБОТКИ ПРЕРЫВАНИЯ ПО GP3
;-----
; Сохранение содержимого регистров STATUS и W в ОЗУ.
;-----
MOVWF W_TEMP ; Копирование содержимого регистра W во временный регистр W_TEMP
SWAPF STATUS,W ; Обмен полубайтов в регистре STATUS с записью в W
MOVWF STATUS_TEMP ; Сохранить STATUS во временном регистре status_temp банка 0

;-----
; Ожидание момента, когда GP3 вернется в '1'
M_GP3_IS_0
BTFSS PORTB,GP3 ; Проверить GP3; - прерывания по изменению уровня сигнала на входах GP,
; пропустить след. инструкцию если 1
GOTO M_GP3_IS_0

;-----
; Сброс флага прерывания (0 бит) регистра intcon
;-----
BTFSC PORTB,GP3 ; Проверить GP3, необходимо для сброса флага GPIF (бит 0 регистра INTCON)
; - прерывания по изменению уровня сигнала на входах GP,
; пропустить след. инструкцию если 0
NOP
BCF INTCON,0 ; Сброс флага GPIF прерывания при любом GP3

;-----
; Восстановление содержимого регистров STATUS и W с последующим выходом из прерывания.
;-----
SWAPF STATUS_TEMP,W ; Обмен полубайтов оригинального значения STATUS
; с записью в W (восстановление текущего банка)
MOVWF STATUS ; Восстановить значение STATUS из регистра W
SWAPF W_TEMP,F ; Обмен полубайтов в регистре W_TEMP с сохранением результата в W_TEMP
SWAPF W_TEMP,W ; Обмен полубайтов в регистре W_TEMP с восстановлением оригинального

```


; значения W без воздействия на STATUS

```
RETfie          ; Возврат из прерывания с разрешением прерываний,  
                ; бит 7 (GIE) регистра INTCON устанавливается в 1  
;=====;  
; КОНЕЦ ОБРАБОТЧИКА ПРЕРЫВАНИЯ  
;=====;  
  
;=====;  
; НАЧАЛО ОСНОВНОЙ ПРОГРАММЫ  
;=====;  
START  
  
; Настройка портов GP0, GP1, GP2 и GP5 на выход  
BSF    STATUS,RPO      ; Установка RPO (5-го бита) регистра STATUS – Выбор банка 1  
MOVLW  b'11011000'    ; Запись константы в регистр W  
MOVWF  TRISB           ; Копирование регистра W в TRISIO (банк 1)  
  
BCF    STATUS,RPO      ; Выбор банка 0  
MOVLW  0x07           ; Выключение компараторов GP0, GP1 и GP2  
MOVWF  CMCON           ; Без выключения компараторов сигнал единицы на GP0 держится всего один цикл  
  
; Разрешение прерывания по входу GP3 на калибровку  
;=====;  
; Для индивидуального разрешения прерываний по изменению уровня сигнала на входах GPIO  
; должны быть глобально разрешены прерывания (GIE= 1) и разрешены прерывания  
; по изменению уровня сигнала на входах (GPIE= 1)  
;=====;  
BSF    STATUS,RPO      ; Выбор банка 1  
BSF    IOCB,IOCB3      ; Разрешение прерывания по изменению уровня сигнала на GP3  
  
BCF    STATUS,RPO      ; Банк 0  
MOVLW  b'00001000'    ; Запись константы в регистр W  
MOVWF  INTCON          ; Копирование W в регистр INTCON, разрешено прерывания по изменению сигнала  
                ; на входах GPIO, бит 3 (GPIE) = 1.  
                ; Глобальное разрешение прерываний заблокировано, бит 7 (GIE) = 0.  
  
MOVLW  1               ; Запись единицы в регистр W необходима для генерации импульсов на GP0 командой XOR  
  
BSF    INTCON,GIE      ; разрешить все прерывания, бит 7 (GIE) = 1.  
;=====;  
; Начало цикла  
;=====;  
LOOP_MAIN  
  
XORWF  PORTB, F;  
BSF    INTCON,GIE      ; разрешить прерывания  
GOTO   LOOP_MAIN  
  
END                ; конец кода программы
```

4. Откомпилируйте программу.

5. Поставьте точку останова в конце основной петли программы (Рис. 33).

```

98 ;=====
99 ; НАЧАЛО ОСНОВНОЙ ПРОГРАММЫ
100 ;=====
101 START
102
103 ; Настройка портов GP0, GP1, GP2 и GP5 на выход
104 BSF STATUS,RPO ; Установка RPO (5-го бита) регистра STATUS - Выбор банка 1
105 MOVLW b'11011000' ; Запись константы в регистр W
106 MOVWF TRISB ; Копирование регистра W в TRISIO (банк 1)
107
108 BCF STATUS,RPO ; Выбор банка 0
109 MOVLW 0x07 ; Выключение компараторов GP0, GP1 и GP2
110 MOVWF CMCON ; Без выключения компараторов сигнал единицы на GP0 держится всего один цикл
111
112 ; Разрешение прерывания по входу GP3 на калибровку
113 ;=====
114 ; Для индивидуального разрешения прерываний по изменению уровня сигнала на входах GP0
115 ; должны быть глобально разрешены прерывания (GIE= 1) и разрешены прерывания
116 ; по изменению уровня сигнала на входах (CPIE= 1)
117 ;=====
118 BSF STATUS,RPO ; Выбор банка 1
119 BSF IOCB,IOCB3 ; Разрешение прерывания по изменению уровня сигнала на GP3
120
121 BCF STATUS,RPO ; Банк 0
122 MOVLW b'00001000' ; Запись константы в регистр W
123 MOVWF INTCON ; Копирование W в регистр INTCON, разрешено прерывания по изменению сигнала
124 ; на входах GP0, бит 3 (CPIE) = 1.
125 ; Глобальное разрешение прерываний заблокировано, бит 7 (GIE) = 0.
126
127 MOVLW 1 ; Запись единицы в регистр W необходима для генерации импульсов на GP0 командой XOR
128
129 BSF INTCON,GIE ; разрешить все прерывания, бит 7 (GIE) = 1.
130 ;=====
131 ; Начало цикла
132 ;=====
133 LOOP_MAIN
134 XORWF PORTB, F;
135 BSF INTCON,GIE ; разрешить прерывания
136 GOTO LOOP_MAIN
137
138 END ; конец кода программы

```

Рис. 33. Точка останова в цикле программы.

- Запустите непрерывное выполнение программы (см. команду на Рис) и в окне Logic Analyzer определите время окончания первого цикла (Рис. 34).

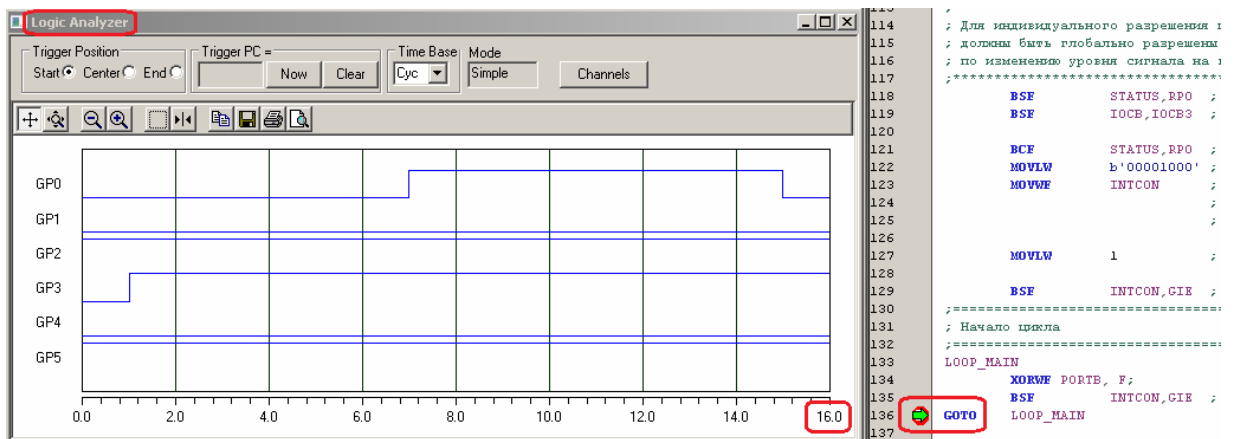


Рис. 34. Определение времени выхода на точку останова.

- Для моделирования входного сигнала на входе GP3 виртуального контроллера откройте окно Stimulus: **Debugger > Stimulus > New Workbook** и укажите времена перехода в ноль и возврата в единицу сигнала GP3 как показано на Рис. 35. Обработчик прерываний запускается если установлено разрешение на прерывание.

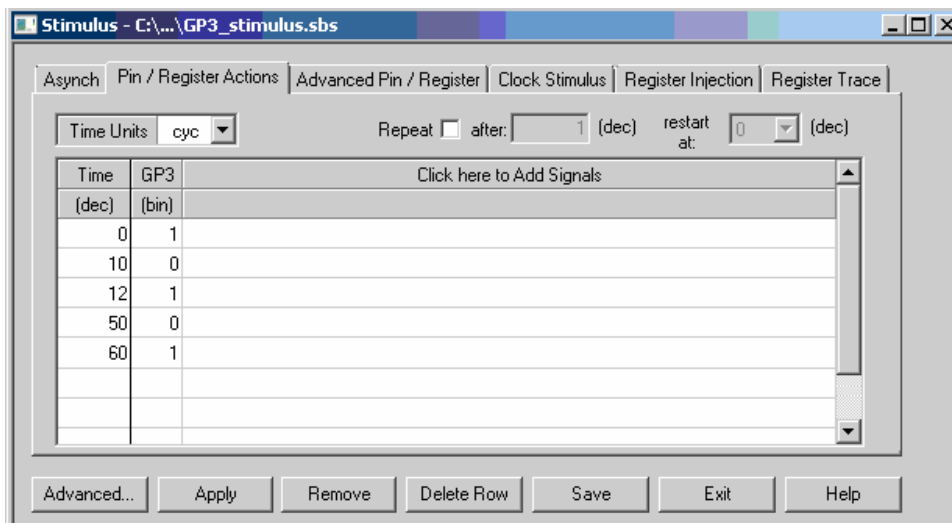


Рис. 35. Пример задания переключений сигнала контроллера на входе GP3.

8. Установите точку останова в теле обработчика прерываний (Рис. 36).

```

56 ;=====
57 ; НАЧАЛО ОБРАБОТКИ ПРЕРЫВАНИЯ ПО GP3
58 ;=====
59 ;
60 ; Сохранение содержимого регистров STATUS и W в ОЗУ.
61 ;-----
62 MOVWF    W_TEMP    ; Копирование содержимого регистра W во временный регистр W_TEMP
63 SWAPF    STATUS,W   ; Обмен полубайтов в регистре STATUS с записью в W
64 MOVWF    STATUS_TEMP ; Сохранить STATUS во временном регистре status_temp банка 0
65 ;-----
66 ;=====

```

Рис. 36. Точка останова в обработчике прерываний.

9. Запуская программу до точек останова и рассмотрите временные диаграммы (Рис. 37). Обратите внимание, что обработка первого прерывания началась после переключения GP3 - в момент выполнения команды

```
BSF    INTCON,GIE    ; разрешить все прерывания, бит 7 (GIE) = 1,
```

а обработка второго прерывания заблокировала генерацию импульсов GP0.

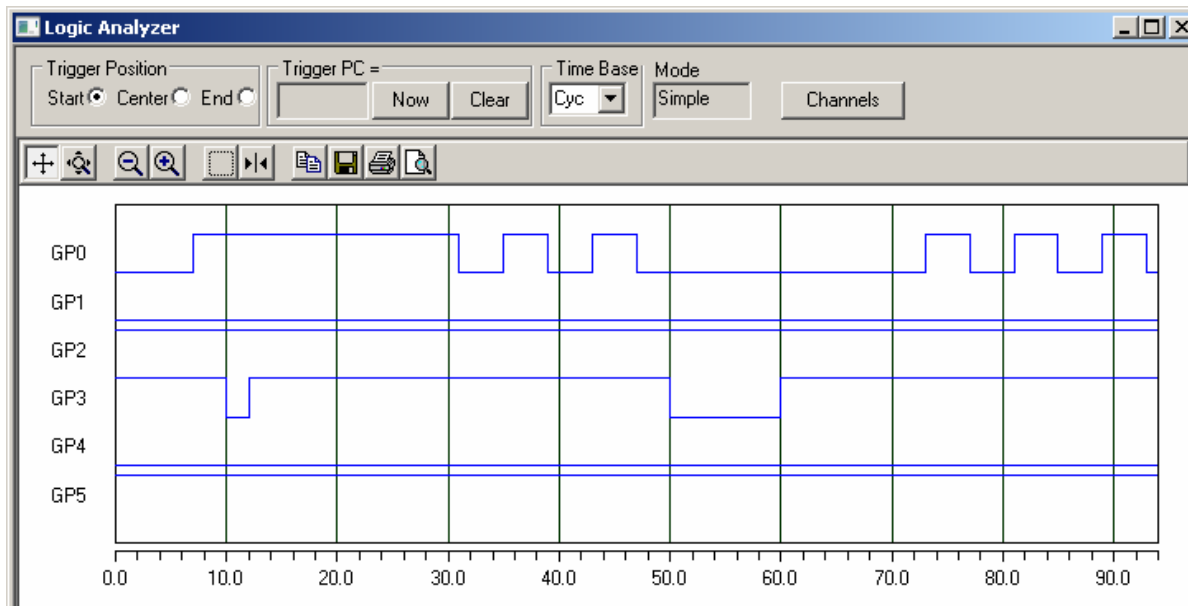


Рис. 37. Временные диаграммы портов контроллера при работе обработчика прерываний.

Задание 3. Компиляция Simulink модели в коды PIC контроллера.

1. Дополните библиотеку Simulink блоками моделирования узлов PIC контроллеров как показано в разделе **Установка компонентов Matlab**.
2. Проверьте результат подключения блоков библиотеки "Embedded Target for Microchip dsPIC" (см. Рис. 13).
3. Установите демонстрационную версию Си компилятора **C30 v3.25**, как показано в соответствующем разделе.
4. Проверьте структуру каталогов установленного компилятора (см. Рис. 15).
5. Выполните **последовательность Simulink программирования для PIC контроллеров**, как показано в одноименном разделе.
6. Проверьте наличие результата - создаваемого HEX файла соответствующей Simulink модели.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какая максимальная частота может быть получена на выходе PIC12F629 контроллера?
2. Каков размер HEX программного файла генератора частоты реализуемого на базе PIC контроллера?
3. Как построить осциллограмму выходного сигнала виртуального контроллера в среде MPLAB?
4. Как смоделировать входные сигналы контроллера в среде MPLAB?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. PIC12F6XX. Однокристальные 8-разрядные FLASH CMOS микроконтроллеры компании Microchip Technology Incorporated. PIC12F629, PIC12F675. ООО "Микро-чип". Москва – 2002. www.microchip.ru.
2. Microchip. MPLAB IDE. User's Guide with MPLAB Editor and MPLAB SIM Simulator http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB_User_Guide_51519c.pdf
3. ICSP Внутрисхемное программирование PIC-контроллеров DOC Rev 1.03 (последнее обновление 19.05.2005) <http://www.5v.ru/icsp.htm>
4. MPLAB IDE Help: MATLAB.
5. Introduction to Microchip-SIMULINK Blocksets and MATLAB Plug-in for MPLAB®IDE Produced by Murali Manohara Chembarpu. http://www.microchip.com/stellent/groups/SiteComm_sg/documents/DeviceDoc/en542925.pdf
6. Embedded Target for 16 bits PIC. <http://www.kerhuel.eu/RTWdsPIC/>
7. PIC12F6XX Однокристальные 8-разрядные FLASH CMOS микроконтроллеры компании Microchip Technology Incorporated • PIC12F629 • PIC12F675. Перевод основывается на технической документации DS4 11 90A компании Microchip Technology Incorporated, USA. <http://www.microchip.ru/files/d-sheets-rus/pic16f62x.pdf>.
8. Dr. Bob Davidov. Портал научно-практических публикаций <http://portalnp.ru/author/bobdavidov>