

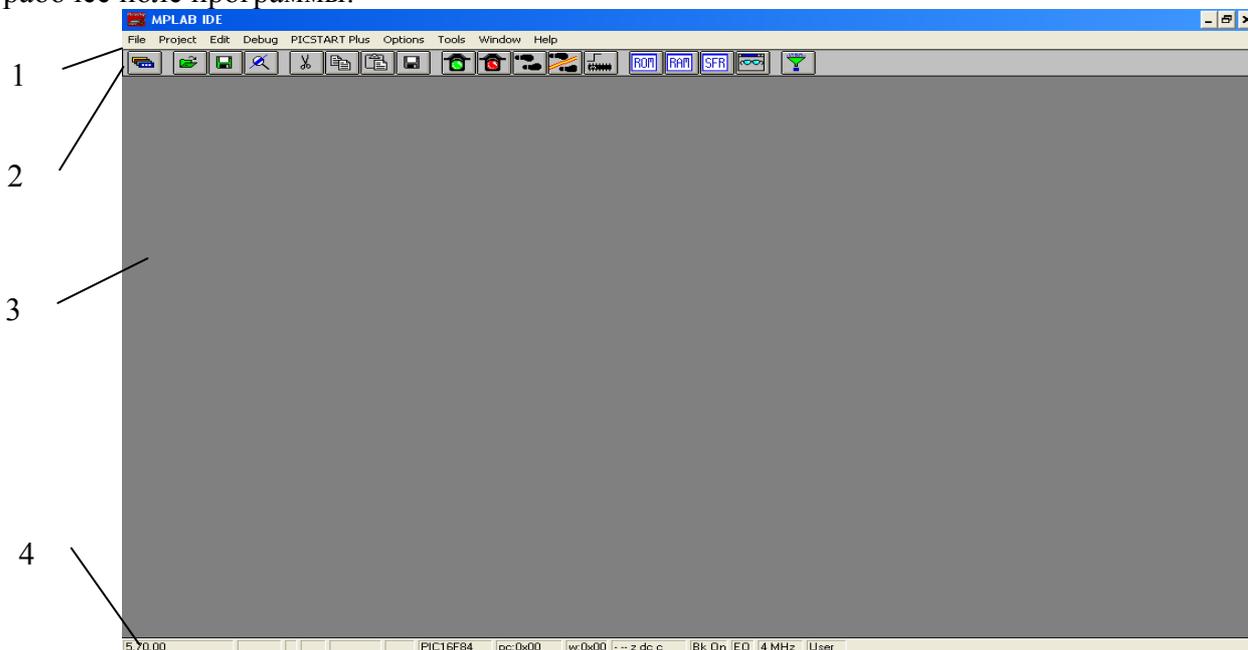
Оглавление

Справочное пособие по изучению пакета MPLAB 5.70.	2
Настройка рабочего проекта.	2
А. Меню пользователя.	6
Список команд МК.....	8
Перечень макроопределений, встроенных в ассемблер MPASM.	9
Регистр STATUS.....	10
Регистр OPTION_REG.....	11
Регистр INTCON.....	12
Управляющие регистры для EEPROM.....	13
Регистр EECON1.....	13
Регистр PCL программный счетчик.....	14
Регистр PORTA.....	14
Регистр TRISA.....	14
Регистр PORTB.....	15
Регистр TRISB.....	15

Справочное пособие по изучению пакета MPLAB 5.70.

Настройка рабочего проекта.

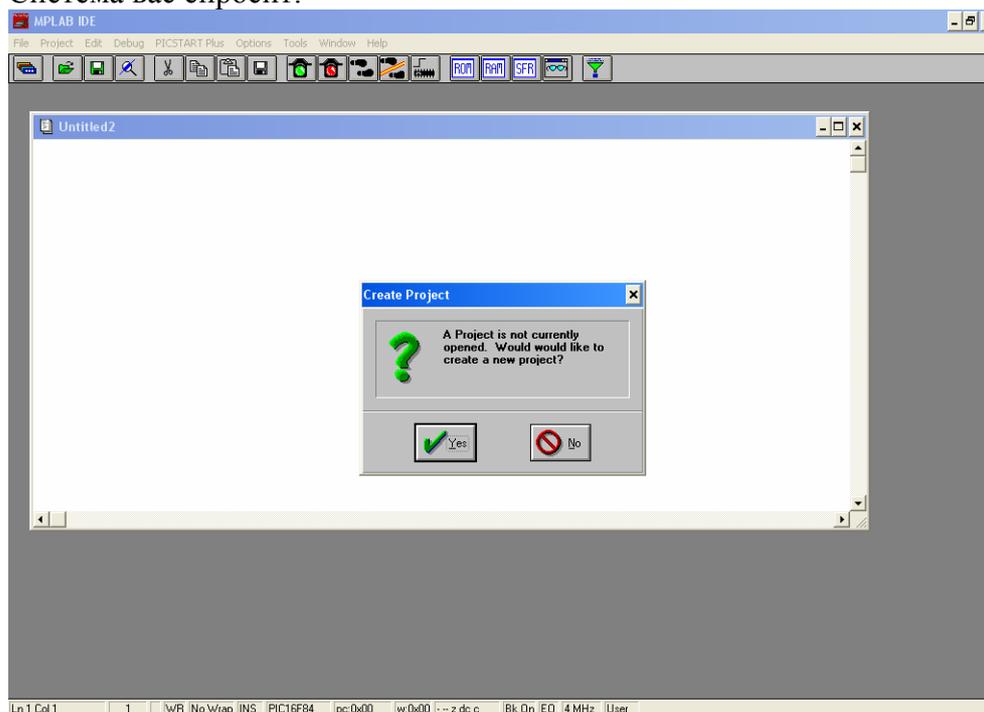
Откройте пакет, запустив файл MPLAB.exe. После запуска вы увидите перед собой рабочее поле программы:



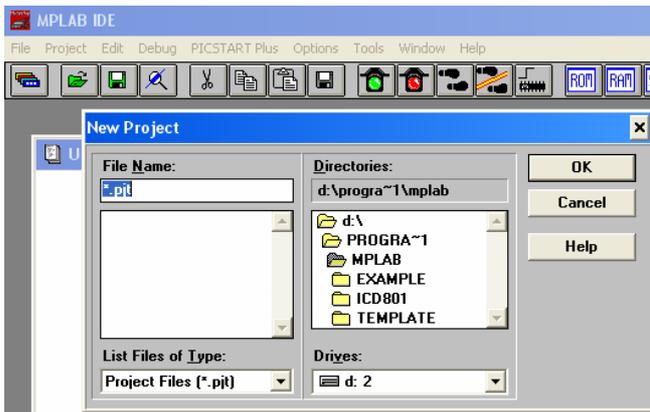
Рабочий стол состоит из:

1. Главное текстовое меню
2. Графическое меню
3. Рабочее поле
4. Линейка состояния, отображающая текущую настройку системы.

Для начала работы необходимо создать проект. Для этого войдите в меню File – New. Система вас спросит:



Нажмите Yes и перейдете к следующему окну:

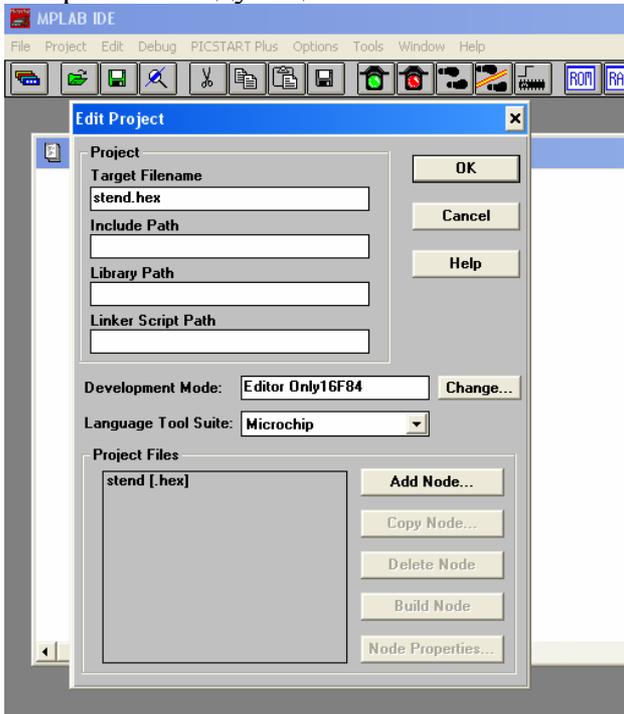


В строчке File Name укажите название проекта, например stend.pjt (не следует писать название русскими буквами и более 8-ми символов). Затем выберите директорию, где будет находиться проект. Необходимо указать следующий каталог:

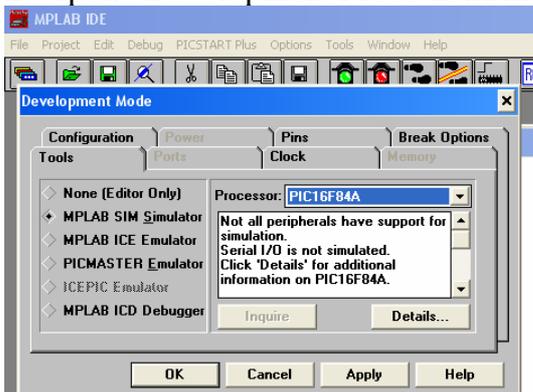
C:\Student\№группы\[\порядковый номер по списку]\[название проекта]. Все имена латинскими буквами и не более 8-ми символов.

Нажмите ОК.

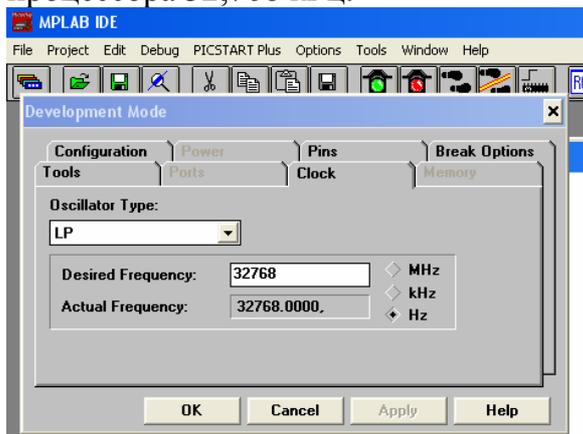
Откроется следующее окно:



В строчке Development Mode нажмите Change.

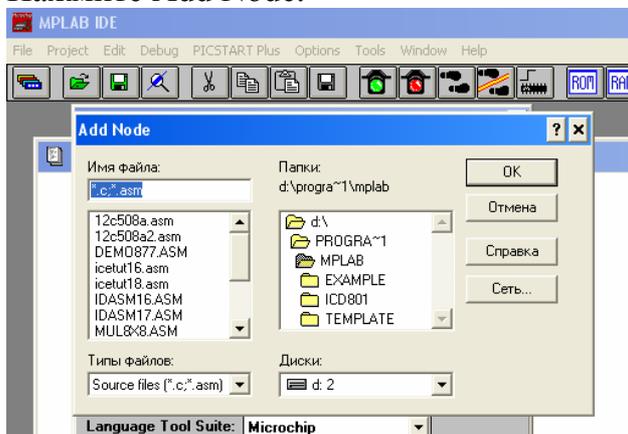


Поставьте ромбик напротив опций MPLAB SIM simulator. В строчке Processor выберите PIC16F84A. Нажмите на вкладку Clock. Установите тактовую частоту процессора 32,768 кГц.

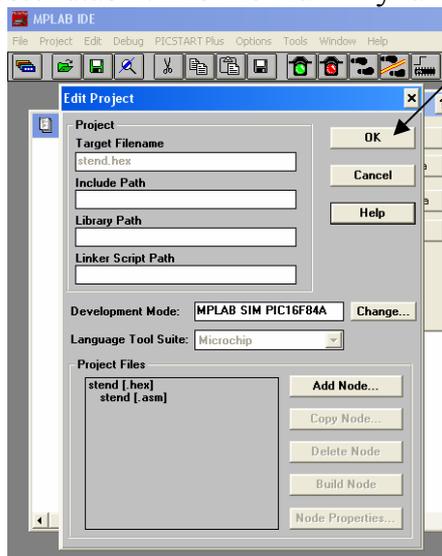


Нажмите ОК. Система выдаст информационные сообщения. Нажимайте ОК, пока не вернетесь в окно Edit project.

Нажмите Add Node.

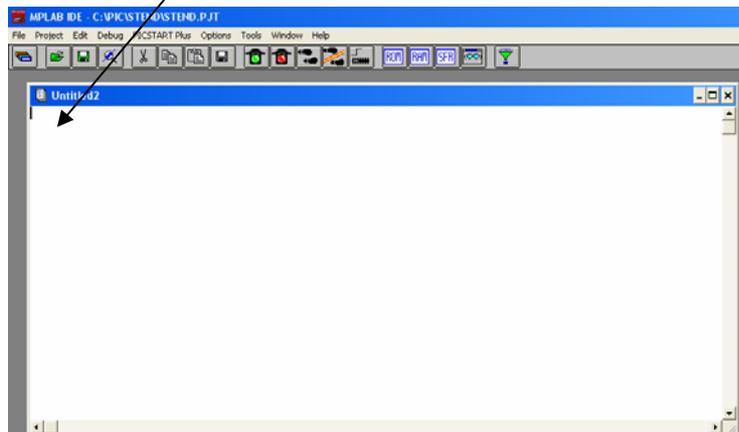


В поле Имя Файла введите имя текстового файла, в котором будет находиться исходный текст программы. Имя должно совпадать с именем проекта. Например stend.asm. В окне Папки укажите путь к вашему проекту. Нажмите ОК.



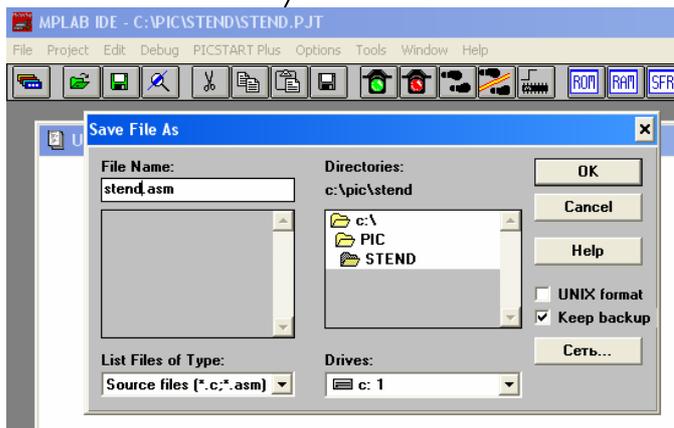
Нажмите ОК. Тем самым вы создадите проект.

Щелкните курсором в начало безымянного текстового файла.

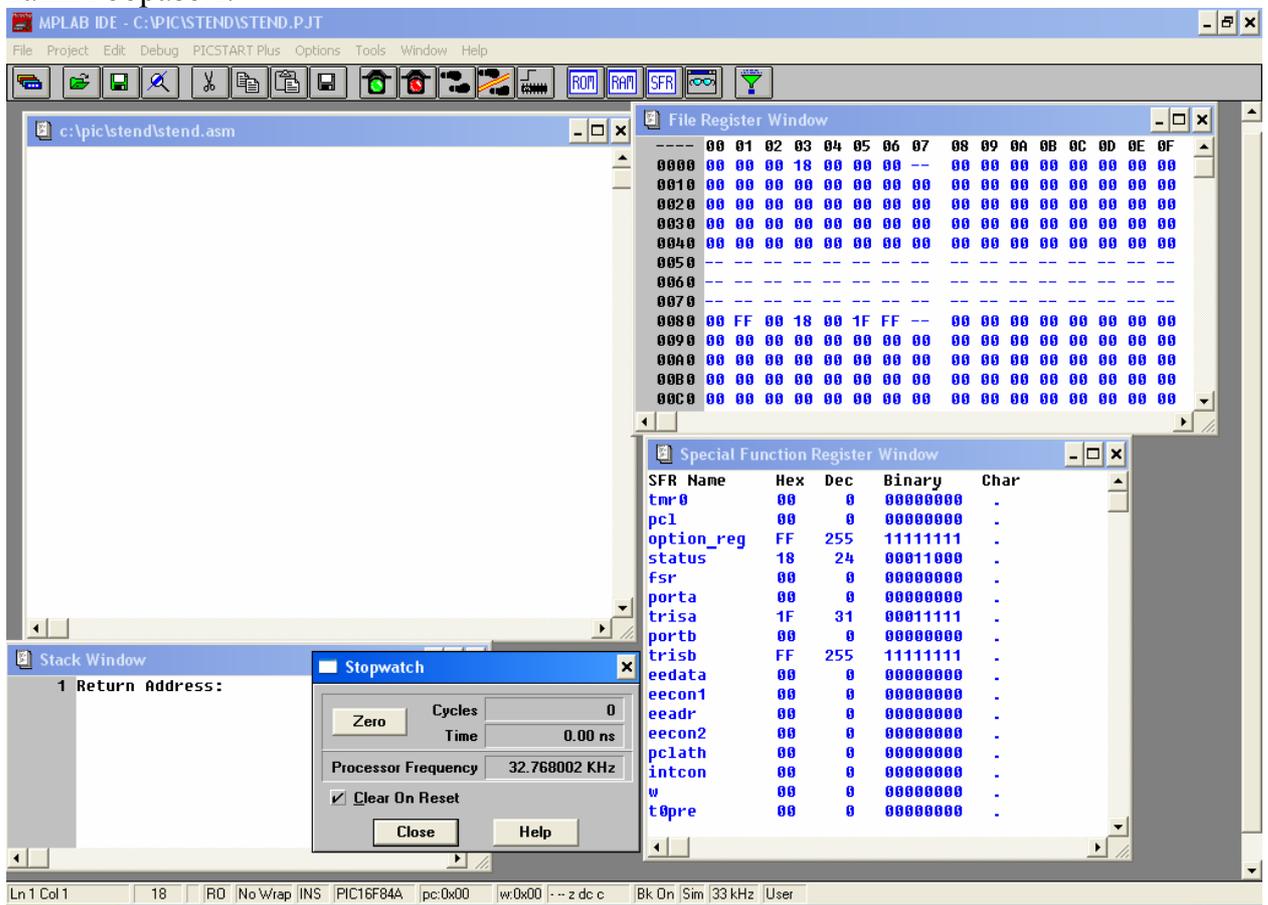


В меню File выберите Save As и сохраните текстовый файл в каталог с проектом, назвав файл по имени проекта. Например Stend.asm. Нажмите ОК. Должно

открыться рабочее поле с текстовым файлом.



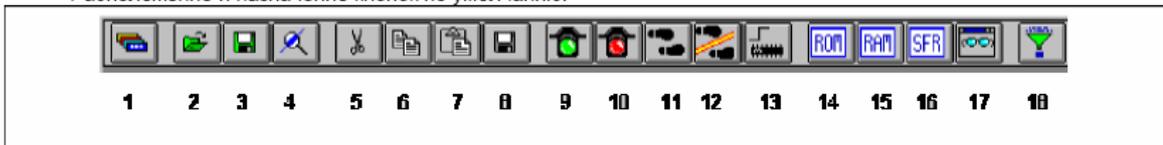
Теперь необходимо открыть окна инструментов. В меню Window находятся: File Register, Special Function Register, Stack, Stopwatch. Расположите окна примерно таким образом:



Теперь сохраните проект, выбрав в меню File – Save All. Настройку можно считать законченной.

A. Меню пользователя.

Расположение и назначение кнопок по умолчанию.



1. Выбор графического меню;
2. Открыть проект;
3. Закреть проект;
4. Найти;
5. Вырезать в буфер;
6. Копировать в буфер;
7. Вставить из буфера;
8. Сохранить файл;
9. Запустить выполнение программы с текущего места;
10. Остановить выполнение программы;
11. Выполнить текущую инструкцию программы;
12. Выполнить текущую инструкцию программы (инструкции CALL выполняются за один шаг с полным выполнением подпрограммы);
13. Сброс микроконтроллера;
14. Окно памяти программ;
15. Окно памяти данных;
16. Окно регистров специального назначения;
17. Новое окно с переменными;
18. Выполнить проект.

В. Назначение полей линейки состояния

Ln 100 Col 2	2048	RO	No Wrap	INS	PIC16C622	pc:0x58	w:0x00	-- z dc c	Bk On	Em	0	Debug
--------------	------	----	---------	-----	-----------	---------	--------	-----------	-------	----	---	-------

Формат поля	Пример отображения	Описание	Результат двойного нажатия левой кнопкой «мыши»
Строка №, Колонка № в активном окне или Версия MPLAB IDE	Ln 1 Col 1 5.00.00	Номер строки и колонки размещения курсора в активном окне редактора Номер версии MPLAB IDE	Открытие диалога перехода на определенную строку Не активно
Строк в файле	72	Отображает количество строк в активном окне редактора MPLAB IDE	Не активно
Модификация файла	#	Символ «#» появляется, если редактируемый файл не был сохранен	Не активно
Защита от редактирования	WR	Включение/отключение защиты от редактирования: WR – редактирование разрешено RD – только чтение	Включает/отключает защиту от редактирования
Выравнивание текста	No Warp	Отображаются параметры выравнивания текста No Warp – выравнивание нет WR 72 – выравнивание текста, максимальное число символов в линии 72	Включает/выключает выравнивание текста
Параметр ввода текста	INS	Отображается параметр ввода текста INS – вставка OVR - замещение	Переключает параметр ввода текста вставка/замещение
Тип микроконтроллера	PIC16C61	Отображается тип микроконтроллера используемого в проекте	Не активно
Счетчик команд	PC:0x5f	Текущее значение счетчика команд PC	Открывает диалог выбора значения счетчика команд
Регистр W	W:0x00	Значение регистра W	Не активно
Биты статуса	ov Z dc c	Состояние битов ALU Верхний регистр – бит установлен Нижний регистр – бит сброшен	Не активно
Флаг разрешения прерываний	Bk On	Глобальное разрешение прерываний	Разрешает/запрещает прерывания
Среда проектирования	Sim	EO – только редактор Sim – симулятор MPLAB SIM Si – симулятор SIMICE ICE – эмулятор MPLAB-ICE Em – эмулятор PICMASTER	Открывает окно настройки проекта
Частота	4MHz	Тактовая частота микроконтроллера	Открывает окно настройки проекта
Режим	Edit	Режим среды проектирования MPLAB IDE	Не активно

Список команд МК

Мнемо-код	Операнды	Выполняемые действия	Изменяемые разряды регистра STATUS
ADDLW	k	Сумма восьмиразрядной константы k и содержимого регистра W заносится в W	C, DC, Z
ADDWF	f, d	Сумма содержимого регистров W и f заносится в W (d=0) или f (d=1)	C, DC, Z
ANDLW	k	Результат поразрядного И содержимого регистра W и восьмиразрядной константы k заносится в W	Z
ANDWF	f, d	Результат поразрядного И содержимого регистров W и f заносится в W (d=0) или f (d=1)	Z
BCF	f, b	В разряд b регистра f заносится лог 0	Нет
BSF	f, b	В разряд b регистра f заносится лог 1	Нет
BTFSC	f, b	Если значение разряда b в регистре f — 0, вместо следующей команды выполняется NOP	Нет
BTFSS	f, b	Если значение разряда b в регистре f — 1, вместо следующей команды выполняется NOP	Нет
CALL	k	Переход к подпрограмме по адресу k. В аппаратный стек заносится адрес следующей за CALL команды, в программный счетчик — константа k	Нет
CLRF	f	Во все разряды регистра f заносится лог 0	Z
CLRWF	Нет	Во все разряды регистра W заносится лог 0	Z
CLRWDI	Нет	Сброс WDT и предварительного делителя. В разряды \overline{TO} и \overline{PD} регистра STATUS заносится лог 1	\overline{TO} , \overline{PD}
COMF	f, d	Все разряды регистра f инвертируются и заносится в W (d=0) или f (d=1)	Z
DECFSZ	f, d	Содержимое регистра f, уменьшенное на 1, заносится в W (d=0) или f (d=1)	Z
DECF	f, d	Аналогичны команде DECF. Если результат нулевой, вместо следующей команды выполняется NOP	Нет
GOTO	k	Безусловный переход по адресу k. В программный счетчик заносится константа k	Нет
INCF	f, d	Содержимое регистра f, увеличенное на 1, заносится в W (d=0) или f (d=1)	Z
INCFSZ	f, d	Аналогичны команде INCF. Если результат нулевой, вместо следующей команды выполняется NOP	Нет
IORLW	k	Результат поразрядного ИЛИ содержимого регистра W и восьмиразрядной константы k заносится в W	Z
IORWF	f, d	Результат поразрядного ИЛИ содержимого регистров W и f заносится в W (d=0) или f (d=1)	Z
MOVF	f, d	Содержимое регистра f заносится в W (d=0) или обратно в f (d=1). Последний вариант служит для проверки содержимого регистра f на 0	Z
MOVLW	k	Восьмиразрядная константа k заносится в регистр W	Нет
MOVWF	f	Содержимое регистра W заносится в регистр f	Нет
NOP	Нет	Переход к следующей команде программы без выполнения каких-либо действий	Нет
OPTION	Нет	Содержимое регистра W заносится в регистр OPTION	Нет
RETFIE	Нет	Возврат из прерывания. В программный счетчик заносится адрес из аппаратного стека, а в разряд GIE регистра INTCON — лог 1	Нет
RETLW	k	Возврат из подпрограммы с кодом завершения. В программный счетчик заносится адрес из аппаратного стека, а в регистр W — восьмиразрядная константа k	Нет
RETURN	Нет	Возврат из подпрограммы. В программный счетчик заносится адрес из аппаратного стека	Нет
RLF	f, d	Содержимое регистра f сдвигается на один разряд в сторону старшего (влево), причем младший разряд принимает значение разряда C рег. STATUS, а старое значение старшего разряда переносится в C	C
RRF	f, d	Содержимое регистра f сдвигается на один разряд в сторону младшего (вправо), причем старший разряд принимает значение разряда C регистра STATUS, а старое значение мл. разряда переносится в C	C
SLEEP	Нет	Перевод МК в "спящий" режим. В разряд \overline{PD} регистра STATUS заносится лог 0, а в разряд \overline{TO} — лог 1. WDT и предварительный делитель сбрасываются. Тактовый генератор выключается	\overline{TO} , \overline{PD}
SUBLW	k	Разность восьмиразрядной константы k и содержимого регистра W заносится в W	C, DC, Z
SUBWF	f, d	Разность содержимого регистров f и W заносится в W (d=0) или f (d=1)	C, DC, Z
SWAPF	f, d	Старшие четыре разряда регистра f и младшие четыре разряда того же регистра заносится в W (d=0) или f (d=1), поменявшись местами (старшие вместо младших и наоборот)	Нет
TRIS	f	Содержимое регистра W заносится в регистр TRISf	Нет
XORLW	k	Результат поразрядной операции ИСКЛЮЧАЮЩЕЕ ИЛИ нед. содержимым регистра W и восьмиразрядной константой k заносится в W	Z
XORWF	f, d	Результат аналогичной операции над содержимым регистров W и f заносится в W (d=0) или f (d=1)	Z

Примечания: 1. При выполнении команды, модифицирующей содержимое регистров PORTx, исходное значение считается непосредственно с выводов соответствующего порта. Например, если вывод запрограммирован как выходной, но внешним воздействием на нем поддерживается низкий уровень, будет считан лог 0 независимо от значения, записанного ранее в соответствующий разряд регистра. 2. Если к TMR0 подключен предварительный делитель, последний обнуляется в момент записи в регистр TMR0 результата выполнения какой-либо команды. 3. В будущем команды OPTION и TRIS будут исключены, поэтому в новых разработках следует пользоваться прямой записью в соответствующие регистры.

Перечень макроопределений, встроенных в ассемблер MPASM.

Макроопределение			Действие	Макроопределение			Действие
ADDCF	macro btfsc incf endm	f,d STATUS,C f,d	Добавить к содержимому регистра f значение признака переноса	MOVFW	macro movf endm	f f,0	Копировать содержимое регистра f в регистр W
ADDDCF	macro btfsc incf endm	f,d STATUS,DC f,d	Добавить к содержимому регистра f значение признака вспомогательного переноса	NEGF	macro comf incf endm	f,d f,1 f,d	Изменить знак числа, находящегося в регистре f
B	macro goto endm	k k	Безусловный переход	SETC	macro bsf endm	STATUS,C	Установить признак переноса
BC	macro btfsc goto endm	k STATUS,C k	Переход, если перенос	SETDC	macro bsf endm	STATUS,DC	Установить признак вспомогательного переноса
BDC	macro btfsc goto endm	k STATUS,DC k	Переход, если вспомогательный перенос	SETZ	macro bsf endm	STATUS,Z	Установить признак нуля
BNC	macro btfss goto endm	k STATUS,C k	Переход, если нет переноса	SKPC	macro btfss endm	STATUS,C	Пропустить следующую команду, если перенос
BNDC	macro btfss goto endm	k STATUS,DC k	Переход, если вспомогательный перенос	SKPDC	macro btfss endm	STATUS,DC	Пропустить следующую команду, если вспомогательный перенос
BNZ	macro btfss goto endm	k STATUS,Z k	Переход, если не ноль	SKPNC	macro btfsc endm	STATUS,C	Пропустить следующую команду, если нет переноса
BZ	macro btfsc goto endm	k STATUS,Z k	Переход, если ноль	SKPNDC	macro btfsc endm	STATUS,DC	Пропустить следующую команду, если нет вспомогательного переноса
CLRC	macro bcf endm	STATUS,C	Снять признак переноса	SKPNZ	macro btfsc endm	STATUS,Z	Пропустить следующую команду, если нет нуля
CLRDC	macro bcf endm	STATUS,DC	Снять признак вспомогательного переноса	SKPZ	macro btfss endm	STATUS,Z	Пропустить следующую команду, если ноль
CLRZ	macro bcf endm	STATUS,Z	Снять признак нуля	SUBCF	macro btfsc decf	f,d STATUS,C f,d	Вычесть из содержимого регистра f значение признака переноса
				SUBDCF	macro btfsc decf endm	f,d STATUS,DC f,d	Вычесть из содержимого регистра f значение признака вспомогательного переноса
				TSTF	macro movf endm	f f,1	Проверить на ноль содержимое регистра f

Подробная информация о SFR регистрах

Регистр STATUS

03h, 83h – STATUS регистр. Регистр содержит арифметические флаги АЛУ, состояние контроллера при сбросе и биты выбора страниц для памяти данных.

STATUS REGISTER (ADDRESS 03h, 83h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x	
IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	
bit 7								bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

C - Флаг переноса/заема:

Для команд ADDWF и SUBWF. Этот бит устанавливается, если в результате операции из самого старшего разряда происходит перенос. Вычитание осуществляется путем прибавления дополнительного кода второго операнда. При выполнении команд сдвига этот бит всегда загружается из младшего или старшего бита сдвигаемого источника.

;-----Пример

```
;SUBWF Example #1
```

```
clrf 0x20 ;f(20h)=0
```

```
movlf 1 ;wreg=1
```

```
subwf 0x20 ;f(20h)=f(20h)-wreg=0-1=FFh
```

```
;Carry=0: Результат отрицательный.
```

```
;SUBWF Example #2
```

```
movlw 0xFF
```

```
movwf 0x20 ;f(20h)=FFh
```

```
clrw ;wreg=0
```

```
subwf 0x20 ;f(20h)=f(20h)-wreg=FFh-0=FFh
```

```
;Carry=1: Результат положительный.
```

1 - перенос

0 – нет переноса

DC - Флаг десятичного переноса/заема:

Для команд ADDWF и SUBWF. Этот бит устанавливается, если в результате операции из четвертого младшего разряда происходит перенос. Механизм установки десятичного бита переноса "DC" тот же самый, отличается тем, что отслеживается перенос из четвертого бита.

1 - перенос

0 – нет переноса

Z - Флаг нулевого результата:

Устанавливается, если результатом арифметической или логической операции является ноль.

PD - Power_Down (режим хранения данных) :

Устанавливается в "1" при включении питания или команде CLRWDT. Сбрасывается в "0" командой SLEEP. Не может быть изменен программно.

TO - Time Out. Флаг срабатывания сторожевого таймера:
 Устанавливается в "1" при включению питания и командами CLRWDT, SLEEP.
 Сбрасывается в "0" по завершению выдержки времени таймера WDT. Не может быть изменен программно.

RP1, RP0 - Биты выбора страницы памяти данных при прямой_адресации
 RP1,RP0:

- 00= Страница 0 (00h-7Fh)
- 01= Страница 1 (80h-FFh)
- 10= Страница 2 (100h-17Fh)
- 11= Страница 3 (180h-1FFh)

IRP - Бит выбора страницы памяти данных при косвенной_адресации
 IRP0:

- 0= Страницы 0,1 (00h-FFh)
- 1= Страница 2,3 (100h-1FFh)

Этот бит в кристалле PIC16C84 использовать не имеет смысла. Поэтому его можно использовать как бит общего назначения чтения/записи. Однако, надо помнить, что в последующих разработках он будет использоваться.

Регистр OPTION_REG

(адрес 81h) доступен для чтения и записи и содержит различные управляющие биты, которые определяют конфигурацию предварительного делителя, куда он подключен: к TMR0 или WDT (сторожевому таймеру), знак фронта внешнего прерывания INT и внешнего сигнала для T0CKI, подключение активной нагрузки на порту RB.

OPTION REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

Выбор коэффициентов деления предделителя.

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

PSA - Бит, подключающий делитель к: 0 - T0CKI, 1 - WDT

T0SE - Фронт внешнего сигнала T0CKI:

0 - инкремент по положительному фронту на выводе T0CKI

1 - инкремент по отрицательному фронту на выводе T0CKI

T0CS - Источник сигнала для T0CKI

0 - сигнал от внутреннего генератора

1 - Внешний сигнал на выводе RTCC

INTEDG - Фронт сигнала INT:

- 0 - прерывание по отрицательному фронту на выводе INT
- 1 - прерывание по положительному фронту на выводе INT

RBPU - бит подключения подтягивающих резисторов к порту В.

- 1 – не подключены
- 0 – подключены.

Регистр INTCON

INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-x						
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7							bit 0

- внешнее прерывание с вывода RB0/INT,
- прерывание от переполнения счетчика/таймера RTCC,
- прерывание после записи данных в EEPROM
- прерывание от изменения сигналов на выводах порта RB<7:4>.

Все прерывания имеют один и тот же вектор/адрес - 0004h. Однако, в управляющем регистре прерываний INTCON записывается:- от какого именно источника поступил запрос прерывания. Записывается соответствующим битом-флагом. Такое прерывание может быть замаскировано индивидуально или глобально битами в этом же регистре INTCON. Единственным исключением является прерывание по концу записи в EEPROM. Этот флаг находится в другом регистре EECON1.

RBIF -Флаг прерывания от изменения на порту RB.

- Флаг устанавливается, когда сигнал на выводе RB<7:4> изменяется.
- Флаг сбрасывается программным способом.

INTF - Флаг прерывания INT.

- Флаг устанавливается, когда на выводе INT появляется сигнал от внешнего источника прерывания. Сбрасывается программным способом.

TOIF - Флаг прерывания от переполнения TMR0.

- Флаг устанавливается, когда TMR0 переполняется.
- Флаг сбрасывается программным способом.

RBIE - Бит разрешения/запрещения RBIF прерывания.

- RBIE = 0 : запрещает RBIF прерывание
- RBIE = 1 : разрешает RBIF прерывание

INTE - Бит разрешения/запрещения INT прерывания.

- INTE = 0 : запрещает INT прерывание
- INTE = 1 : разрешает INT прерывание

TOIE Бит разрешения/запрещения TOIF прерывания.

- TOIE = 0 : запрещает TOIF прерывание
- TOIE = 1 : разрешает TOIF прерывание

EEIE - Бит разрешения/запрещения прерывания EEPROM записи.

EEIE = 0 : запрещает EEIF прерывание

EEIE = 1 : разрешает EEIF прерывание

GIE Бит разрешения/запрещения всех прерываний.

GIE = 0 : запрещает прерывания

GIE = 1 : разрешает прерывания

Он сбрасывается автоматически при следующих событиях:

- при включении питания.
- по внешнему сигналу /MCLR при нормальной работе.
- по внешнему сигналу /MCLR в режиме SLEEP.
- после задержки таймера WDT при нормальной работе.
- после задержки таймера WDT в режиме SLEEP.

Управляющие регистры для EEPROM

Название	Функция	Адрес	Значение после включения
EEDATA	EEPROM регистр данных	08h	XXXX XXXX
EEADR	EEPROM регистр адреса	09h	XXXX XXXX
EECON1	EEPROM 1 управляющий регистр	88h	0000 X000
EECON2	EEPROM 2 управляющий регистр	89h	-

Регистр EECON1

EECON1 REGISTER (ADDRESS 88h)

U-0	U-0	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0	
—	—	—	EEIF	WRERR	WREN	WR	RD	
bit 7								bit 0

Управляющие биты RD и WR запускают соответственно чтение и запись. Они могут быть установлены только программно. Сбрасываются- аппаратно по завершения операций чтения/записи. Запрет программного сброса бита WR предотвращает преждевременное окончание записи.

+-----+

| RD | - Бит чтения.

+-----+ RD =1 : Запускает чтение памяти данных EEPROM. Чтение занимает один цикл. Устанавливается программно. Обнуляется аппаратно.

+-----+

| WR | - Бит записи.

+-----+ WR =1 : Запускает запись в память данных EEPROM. Устанавливается программно. Обнуляется аппаратно.

+-----+

! WREN ! - Бит разрешения записи в память данных EEPROM.

+-----+ WREN = 1: Разрешена запись.

WREN = 0: Запрещена запись.

После включения питания WREN обнуляется.

Флаг ошибки WRERR устанавливается, когда процесс записи прерывается сигналом сброса /MCLR или сигналом сброса от WDT таймера. Рекомендуем проверять этот флаг WRERR и при необходимости производить перезапись данных, данные и адрес которых сохраняются в регистрах EEDATA и EEADR.

+-----+

!WRERR ! - Флаг ошибки записи.

+-----+ WRERR = 1: Флаг устанавливается, когда операция записи преждевременно прерывается сигналом сброса /MCLR (во время обычного режима или режима SLEEP) или сигналом сброса WDT во время обычного режима.

Флаг EEIF устанавливается, когда встроенный автомат завершает запись в память данных. Он должен быть сброшен программно.

+-----+

! EEIF ! - Флаг завершения записи.

+-----+ EEIF = 1: Флаг устанавливается, когда завершена запись.

Соответствующий бит разрешения прерывания - EEIE в регистре INTCON.

Регистр PCL программный счетчик

Ширина программного счетчика - 13 бит. Младший байт программного счетчика PCL доступен для чтения и записи и находится в регистре 02h. Старший байт программного счетчика (PCH) не может быть прямым образом прочитан или записан. Старший байт программного счетчика защелкивается из PCLATH регистра, адрес которого 0Ah. В зависимости от того, загружается ли в программный счетчик новое значение во время выполнения команд CALL, GOTO, или в младший байт программного счетчика PCL производится запись, - старшие биты программного счетчика выгружаются из PCLATH разными способами.

Регистр PORTA

Порт А - это порт шириной 5 бит, ему соответствуют выводы кристалла RA<4:0>. Линии RA<3:0> двунаправленные, а линия RA4 - выход с открытым стоком. Адрес регистра порта А - 05h. Относящийся к порту А управляющий регистр TRISA расположен на первой странице регистров по адресу 85h. TRISA<4:0> - это регистр шириной 5 бит. Если бит управляющего TRISA регистра имеет значение единица, то соответствующая линия будет устанавливаться на вход. Ноль переключает линию на выход и одновременно выводит на нее содержимое соответствующего регистра защелки. Ниже дана схема портов RA0..RA3

Регистр TRISA

Управляющий регистр порта А

Настройка производится путем записи управляющих слов в TRIS регистры.

Если бит управляющего TRISA регистра имеет значение единица, то соответствующая линия PORTA будет настроена на вход. Ноль переключает линию на выход.

Регистр PORTB

Порт В - это двунаправленный порт, шириной в восемь бит (адрес регистра 06h). Относящийся к порту В управляющий регистр TRISB расположен на первой странице регистров по адресу 86h. Если бит управляющего TRISB регистра имеет значение единица, то соответствующая линия будет устанавливаться на вход. Ноль переключает линию на выход и одновременно выводит на нее содержимое соответствующего регистра защелки.

У каждого вывода порта В имеется небольшая активная нагрузка (около 100мкА) на линию питания Vdd. Она автоматически отключается, если этот вывод запрограммирован как выход. Более того, управляющий бит RBPU OPTION<7> может отключить (RBPU=1) все нагрузки. Сброс при включении питания также отключает все нагрузки.

Четыре линии порта RB<7:4> имеют способность вызывать прерывание при изменении значения сигнала на любой из них. Если эти линии настроены на вход, то они опрашиваются и защелкиваются в цикле чтения Q1:. Новая величина входного сигнала сравнивается со старой в каждом командном цикле. При несовпадении значения сигнала на выводе и в защелке, генерируется высокий уровень. Выходы детекторов "несовпадений" RB4,RB5,RB6,RB7 объединяются по ИЛИ и генерируют прерывание RBIF . Любая линия, настроенная как выход, не участвует в этом сравнении. Прерывание может вывести кристалл из режима SLEEP. В подпрограмме обработки прерывания следует сбросить запрос прерывания одним из следующих способов:

- 1) Запретить прерывания при помощи обнуления бита RBIE INTCON<3>.
- 2) Прочитать порт В. Это завершит состояние сравнения.
- 3) Обнулить бит RBIF INTCON<0>.

Прерывание по несовпадению и программно устанавливаемые внутренние активные нагрузки на этих четырех линиях могут обеспечить простой интерфейс, например, с клавиатурой, с выходом из режима SLEEP по нажатию клавиш.

Вывод RB0 совмещен со входом внешнего прерывания INT.

Регистр TRISB

Управляющий регистр порта В

Настройка производится путем записи управляющих слов в TRISB регистры.

Если бит управляющего TRISB регистра имеет значение единица, то соответствующая линия PORTA будет настроена на вход. Ноль переключает линию на выход.