



## Описание языка Бейсик ЭпплеСофт

### Точки входа во внутренние подпрограммы

Crossley - Apple Orchard Mar/Apr 1980

Это примерно то, что мы искали и не нашли в своё время ...

Перевод на русский язык Серков С.В. (Serkov S.V) 2025 (через 45 лет !)

все права на данный перевод принадлежат автору перевода

Document # 43  
Ex Libris David T. Craig

**Внутренние точки входа Applesoft**

Автор: Джон Кроссли (из Apple Orchard)

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ	51
МЕТКИ	52
СОКРАЩЕНИЯ	51
ВХОДНЫЕ ПРОЦЕДУРЫ ТХТР	52
ПРОЦЕДУРЫ ТХТР В ЦЕЛЫЕ ЧИСЛА	52
ПАКЕТ МАТЕМАТИКИ С ПЛАВАЮЩЕЙ ТОЧКОЙ	
ВВЕДЕНИЕ	52
РЕГИСТРЫ	52
ОПЕРАТОРЫ	52
КОНСТАНТЫ	52
ФУНКЦИИ	53
ПРОЦЕДУРЫ ПЕРЕМЕЩЕНИЯ	53
УТИЛИТЫ	53
ПРЕОБРАЗОВАНИЯ	
ЦЕЛОЕ ЧИСЛО В FAS	53
FAS В ЦЕЛОЕ ЧИСЛО	53
ТХТР В FAS	53
УТИЛИТЫ СТРОК	54
УСТРОЙСТВО ВХОДНЫЕ ПРОЦЕДУРЫ	54
ПРОЦЕДУРЫ ВЫХОДА УСТРОЙСТВА	54
ПРОЦЕДУРЫ ВНУТРЕННЕГО ЛОКатора	55
ПРОЦЕДУРЫ ИНИЦИАЛИЗАЦИИ	55
ПРОЦЕДУРЫ УПРАВЛЕНИЯ ХРАНИЛИЩЕМ	55
РАЗЛИЧНЫЕ ОСНОВНЫЕ КОМАНДЫ	55
ПРОЦЕДУРЫ ГРАФИКИ ВЫСОКОГО РАЗРЕШЕНИЯ	55
ПРОЦЕДУРЫ КАССЕТЫ	56
ПРОЦЕДУРЫ ОБРАБОТКИ ОШИБОК	56
ПРОЦЕДУРЫ ПРОВЕРКИ СИНТАКСИСА	56
ИНДЕКС	57

**ВВЕДЕНИЕ**

Это руководство для программиста машинного языка MS 6502, по использованию различных подпрограмм Applesoft. Указанные адреса подпрограмм предполагают, что у пользователя ПК Apple II Plus, карта прошивки Applesoft или "Языковая карта" (Language Card). Этот список считается правильным, но имейте в виду, что описание было сделано в свободное от основной работы время. Если вы обнаружите ошибки, свяжитесь с вашей группой пользователей. Эти данные предназначены для опытных программистов, А НЕ ДЛЯ НОВИЧКОВ. Прочитайте справочное руководство Applesoft для получения дополнительной информации.

Обратите особое внимание на подпрограмму CHRGET. Эта подпрограмма является сердцем языка Basic Applesoft. Когда Applesoft требуется следующий символ или инструкция, он указывает ТХТР на программу или входной буфер и выполняет JSR на CHRGET.

Когда Applesoft читает данные оператора DATA, ТХТР временно устанавливается на последний использованный оператор DATA

**МЕТКИ НУЛЕВОЙ СТРАНИЦЫ И ИХ АДРЕСА**

A1	3C,3D	Указатель монитора Apple для подпрограмм чтения/записи на кассету
A2	3E,3F	Указатель монитора Apple для подпрограмм чтения/записи на кассету
ARYTAB	6B,6C	Адрес начала области хранения массивов
BUF	200,2FF	Буфер ввода строки
CHARAC	0D	Используется STRLT2
CURLIN	75,76	Текущий номер строки выполняемой Бейсик-программы (=FF, если Бейсик в режиме диалога)
DATLIN	7B,7C	Номер строки программы оператора DATA, текущее чтение которых осуществляется READ
DATPTR	7D,7E	Адрес, с которого поступают следующие данные DATA, читаемые READ
DSCTMP	9D,9E,9F	Временный строковый дескриптор (при работе со строками)
ENDCHR	0E	Используется SRTL2
ERRFLG	D8	\$80, если ONERR активен
ERRLIN	DA,DB	Номер строки программы, в которой произошла ошибка
ERRNUM	DE	Код ошибки
ERRPOS	DC,DD	TXTPTR сохраненное для HNDLERR (адрес до адреса оператора, при выполнении которого возникла ошибка)
ERRSTK	DF	Значение указателя стека до ошибки
FBUFR	100-110	FOUT буфер (стек)
FIRST	F0	Используется PLOTFS
FORPNT	85,86	Общий указатель, см. COPY
FRESC	71,72	Временный указатель для подпрограмм хранения строк
FRETOP	6F,70	Нижняя граница памяти для хранения текстовых переменных
H2	2C	Используется PLOITNS
HIGHDS	94,95	Используется BLTU
HIGHTR	96,97	Используется BLTU
HPAG	E6	HIRES страница для графики высокого разрешения (\$20 для HGR, \$40 для HGR2)
INDEX	5E,5F	Временный указатель для перемещения строк
INVFLG	32	Маска для обратного вывода (цвет символа, выводимого на экран в формате дисплейного контроллера)
LASTPT	53	Последний использованный временный указатель строки
LINNUM	50,51	Расположение 16-битного числа общего назначения
LOWTR	9B,9C	Регистр общего назначения. Используется GETARIPT FINDLN, BLTU
MEMSIZ	73,74	Адрес конца ОЗУ (верхняя граница памяти, HIMEM)
OLDLIN	77,78	Последняя выполненная строка
ORMASK	F3	Маска для мигающего вывода
PRGEND	AF,B0	Адрес конца текста программы пользователя
REMSTK	F8	Указатель стека сохраняется перед каждым оператором
ROT	F9	Указатель поворота фигуры для программ графики
SCALE	E7	Масштаб фигуры для программ графики
SPDBYT	F1	Значение скорости вывода информации в обратном коде. Устанавливается командой SPEED.
STREND	6D,6E	Адрес верхней границы памяти хранения массивов
STRNG1	AB,AC	Указатель на строку. См. MOVINS
STRNG2	AD,AE	Указатель на строку. См. STRLT2
SUBFLG	14	\$00 индексы разрешены, \$80=нет индексов
TEMPPT	52	Последний использованный временный строковый дескриптор
TXTPTR	B8,B9	Следующий байт для чтения строки программы или данных (CHRGET)
TXTTAB	67,68	Адрес начала хранения текста программы
V2	2D	Используется PLOTFS (знак курсора, обычно \$DE)
VALTYP	11	Флаг типа последней использованной переменной или операции FAC 0=число, FF= строка
VARPNT	83,84	Адрес указателя к значению последней, использованной переменной. Используется PTRGET
VARTAB	69,6A	Адрес начала области хранения переменных (LOMEM)

## Точки входа во внутренние подпрограммы языка Бейсик ЭпплСофт

### Обозначения

A - аккумулятор процессора 6502, X - регистр X, Y - регистр Y

Z - нулевой флаг регистра состояния процессора 6502

C - флаг переноса регистра состояния 6502

A,X — это 16-битное число, где A — старший байт, а X — младший байт.

(Y,A) — это число или строка, адрес которой находится в Y и A, где старший бит находится в Y, а первый бит — в A.

FAC - аккумулятор чисел с плавающей точкой (запятой)

ARG - регистр аргумента при вычислениях в формате с плавающей запятой

msb - старший бит или байт

lsb - младший бит или байт

eol - токен конца строки (\$00)

## TXTPTR ВХОДНЫЕ ПРОЦЕДУРЫ подпрограммы чтения строк программы/данных

CHRGET 00B1 (177) (читает символ строки увеличивая TXTPTR)

CHRGOT 00B7(183) (без увеличения TXTPTR)

Эти процедуры загружают символ в A из TXTPTR и устанавливают определенные флаги состояния 6502.

Регистры X и Y не изменяются.

На выходе: A=прочитанный символ

Z устанавливается, если A — это «:» или eol (\$3A или \$00)

C очищается, если A — это число ASCII (от 0 до 9).

## TXTPTR В ЦЕЛОЕ ЧИСЛО

### LINGET DA0C (55820)

Прочитать номер строки (целое число до 63999) из TXTPTR в LINNUM. LINGET предполагает, что регистры 6502 и A были установлены CHRGET, который извлек первую цифру. Обычно выходит через CHRGET, который извлекает символ после числа. Если число больше 63999, то LINGET выходит через SYNTAX ERROR. LINNUM равен нулю, если в TXTPTR нет числа.

### GTBYTC E6F5 (59125)

Обращается к CHRGET, чтобы пропустить символ и перейти к GETBYT.

### GETBYT E6F8 (59128)

Вычисляет формулу, на которую указывает TXTPTR, оставляет результат в FAC и попадает в CONINT. На входе TXTPTR должен указывать на первый символ формулы.

### PLOTFNS F1EC (61932)

Получить 2 координаты построения LO-RES (графики низкого разрешения) (0-47,0-47) из TXTPTR, разделенные запятой. При вводе TXTPTR указывает на первый символ формулы для первого числа. PLOTFNS помещает первое число в FIRST и второе число в H2 и V2.

### HFNS F6B9 (63161)

Получить координаты построения HI-RES (графики высокого разрешения) (0-279,0-191) из TXTPTR. При вводе TXTPTR указывает на первый символ формулы для первого числа. Оставляет регистры 6502 настроенными для HPOSN.

На выходе:

A=вертикальная координата, X=младший бит горизонтальной координаты

Y=старший бит горизонтальной координаты.

## ПАКЕТ МАТЕМАТИКИ С ПЛАВАЮЩЕЙ ТОЧКОЙ

### ВВЕДЕНИЕ

Числа с плавающей точкой (запятой) это числовой формат, используемый всюду в Basic Applesoft:  
 Экспонента — это однобайтовое число со знаком (EXP) в форме, превышающей \$80 (к знаковому значению добавлено \$80). Мантисса — это 4 байта (НО, МОН, МО, LO). Предполагается, что двоичная точка находится справа от самого старшего бита. Поскольку в двоичной нотации с плавающей точкой старший бит всегда равен 1, знак числа хранится там, когда число хранится в упакованном виде в памяти. В то время как в математическом пакете знак хранится в отдельном байте (SGN), где значимым является только бит 7. Если экспонента равна нулю, то число равно нулю, хотя мантисса не обязательно равна нулю.

### Примеры:

EXP NO MOH MO LO SGN

#### Packed format

-10 84 A0 00 00 00

10 84 20 00 00 00

EXP NO MOH MO LO SGN

#### FAC format

-10 84 A0 00 00 00 FF

10 84 A0 00 00 00 00

### Соглашения о вызове арифметических процедур:

Для функций с одним аргументом:

Аргумент находится в FAC.

Результат остается в FAC.

Для функций с двумя аргументами:

Первый аргумент находится в ARG (см. CONUPK).

Второй аргумент находится в FAC.

Результат остается в FAC.

## РЕГИСТРЫ С ПЛАВАЮЩЕЙ ТОЧКОЙ

ПРИМЕЧАНИЕ: многие из следующих ячеек памяти используются для других целей, когда не используются пакетом математической обработки с плавающей точкой.

	FAC	ARG	TEMP1	TEMP2	TEMP3	RND
EXP 9D	A5	93	98	8A	C9	
NO 9E	A6	94	99	8B	CA	
MOH 9F	A7	95	9A	8C	CB	
MO A0	A8	96	9B	8D	CC	
LO A1	A9	97	9C	8E	CD	
SGN A2	AA	(компактный формат)				

## ПОДПРОГРАММЫ ВЫЧИСЛЕНИЯ С ПЛАВАЮЩЕЙ ТОЧКОЙ

**FMULT E97F (59775)** Переместить число в памяти, на которое указывает Y,A, в ARG и попасть в ...

**FMULTT E982 (59778)** Перемножить FAC и ARG. На входе A и Z отражают FACEXP.

**FDIV EA66 (60006)** Переместить число в памяти, на которое указывает Y,A в ARG и попасть в ...

**FIDVT EA69 (60009)** Разделите ARG на FAC. На входе A и Z отражают FACEXP.

**FADD E7BE (59326)** Переместить число в памяти, на которое указывает Y,A в ARG и попасть в ...

**FADDT E7C1 (59329)** Сложить FAC и ARG. На входе A и Z отражают FACEXP.

**FSUB E7A7 (59303)** Переместить число в памяти, на которое указывает Y,A в ARG и попасть в ...

**FSUBT E7AA (59306)** Вычесть FAC из ARG. На входе A и Z отражают FACEXP.

**FPWRT EE97 (61079)** Возведение в степень (ARG в FAC). На входе A и Z должны отражать значение FACEXP.

ПРИМЕЧАНИЕ: Большинство процедур перемещения FAC настраивают A и Z для отражения FACEXP, но LDA \$9D обеспечит правильные значения.

## КОНСТАНТЫ С ПЛАВАЮЩЕЙ ТОЧКОЙ

Таблица адресов чисел в упакованной форме, подходящих для использования CONUPK и MOVMF.

RND	00C9	(201)	случайное число
1/4	F070	(61552)	одна четвертая
1/2	EE64	(61028)	одна вторая
-1/2	E937	(59703)	минус одна вторая
1	E913	(59667)	единица
10	EA50	(59984)	десять

**SQR(.5) E92D (59693)** квадратный корень из 0,5  
**SQR(2) E932 (59698)** квадратный корень из 2  
**LN(2) E93C (59708)** натуральный логарифм 2  
**LOG(e)2 EEDB (61147)** логарифм e в степени 2

**PI/2 F063 (61539)** число Пи/2  
**PI\*2 F06B (61547)** число Пи\*2  
**-32768 E0FE (57598)**  
**100000000 ED14[1E9] (60692[489])**

### ФУНКЦИИ С ПЛАВАЮЩЕЙ ТОЧКОЙ

#### SGN EB90 (60304)

Вызывает SIGN и помещает результат в FAC. На выходе: FAC=1 Если FAC был больше 0; FAC=0 Если FAC был равен 0; FAC=-1 Если FAC был меньше 0

**ABS EBAF (60335)** Абсолютное значение FAC

**INT EC23 (60451)** Наибольшее целое значение FAC. Использует QINT и делает результат плавающим.

**SQR EE8D (61069)** Извлечь квадратный корень из FAC

**LOG E941 (59713)** Логарифм по основанию e из FAC

**EXP EF09 (61193)** Возвести e в степень FAC

**RND EF4E (61358)** Сформировать «случайное» число в FAC

**COS EFEA (61418)** COS(FAC)

**SIN EFF1 (61425)** SIN(FAC)

**TAN F03A (61498)** TAN(FAC)

**ATN F09E (61598)** ARCTAN(FAC)

### ЧИСЛА С ПЛАВАЮЩЕЙ ТОЧКОЙ (ПОДПРОГРАММЫ ПЕРЕМЕЩЕНИЯ)

#### MOVFM EAF9 (60153)

Переместить память, на которую указывают Y,A, в FAC. На выходе A и Z отражают FACEXP.

#### MOV2F EB1E (60190)

Упаковывает FAC и перемещает его во временный регистр 2 (TEMP2). Использует MOVFM. На выходе A и Z отражают FACEXP.

#### MOVIF EB21 (60193)

Упаковывает FAC и перемещает его во временный регистр 1 (TEMP1). Использует MOVFM. На выходе A и Z отражают FACEXP.

#### MOVML EB23 (60195)

Упаковывает FAC и перемещает его в область нулевой страницы, на которую указывает X.

Использует MOVFM. На выходе A и Z отражают FACEXP.

#### MOVMF EB2B (60203)

Упаковывает FAC и перемещает его в память, на которую указывают Y, X. На выходе A и Z отражает FACEXP.

#### MOVFA EB53 (60243)

Переместить ARG в FAC. На выходе A=FACEXP и Z установлен.

#### MOVAF EB63 (60259)

Переместить FAC в ARG. На выходе A=FACEXP и Z устанавливается.

#### CONUPK E9E3 (59875)

Загрузить ARG из памяти, на которую указывают Y, A. На выходе A и Z отражают FACEXP.

#### Сводная таблица подпрограмм перемещений:

FAC => (Y,A) EB2B
FAC => (0,X) EB23
FAC => TEMP1 EB21

FAC => TEMP2 EB1E
FAC => ARG EB63
(Y,A) => FAC EAF9

(Y,A) => ARG E9E3
ARG => FAC EB53

**ЧИСЛА В ФОРМАТЕ С ПЛАВАЮЩЕЙ ТОЧКОЙ (УТИЛИТЫ)****SIGN EB82 (60290)**

Установить A в соответствии со значением FAC

На выходе: A=1, если FAC положительный; A=0, если FAC=0; A=FF, если FAC отрицательный

**FOUT ED34 (60724)**

Создает строку в FBUFFR, эквивалентную значению FAC. На выходе Y,A указывает на строку. Строка заканчивается нулем. FAC перемешивается. Используйте STROUT, чтобы затем вывести число.

**FCOMP EBB2 (60338)**

Сравнить FAC и упакованное число в памяти, на которое указывает Y,A.

На выходе: A=1 если (Y,A)<FAC; A=0 если (Y,A)=FAC; A=FF если (Y,A)>FAC

**NEGOP EEDO (61136)** Превратить FAC в -FAC**FADDH E7A0 (59296)** Прибавить 1/2 к FAC**DIVIO EA55 (59989)** Разделить FAC на 10. Возвращает только положительные числа.**MULIO EA39 (59961)** Умножить FAC на 10. Работает как для положительных, так и для отрицательных чисел.**ПРЕОБРАЗОВАНИЯ ЦЕЛОГО ЧИСЛА В FAC****SNGFLT E301 (58113)**

Преобразует беззнаковое целое число в Y в формат с плавающей точкой. Загружает его в FAC.

**GIVAYF E2F2 (58098)**

Преобразует целое число со знаком в A,Y в формат с плавающей точкой. Загружает его в FAC.

**FLOAT EB93 (60307)**

Преобразует целое число со знаком, загружая его из A, в формат с плавающей точкой и загружает его в FAC.

**ПРЕОРАЗОВАНИЯ FAC В ЦЕЛОЕ ЧИСЛО****CONINT E6FB (59131)**

Преобразует FAC в однобайтовое число, возвращая его в регистре X и FACLO. Обычно выходит через CHRGET. Если FAC > 255 или < 0, то CONINT выходит через ошибку "ILLEGAL QUANTITY ERROR" (ОШИБОЧНОЕ ЗНАЧЕНИЕ).

**AYINT E10C (57612)** Если FAC меньше +32767 и больше -32767, то выполнит QINT.

**QINT EBF2 (60402)**

Быстрая функция наибольшего целого числа. Оставляет INT (FAC) в FACHO, MO, LO со знаком. QINT принимает FAC до 2 в степени 23 (8388608 в десятичной системе)

**GETADR E752 (59218)**

Преобразует число в FAC (-65535 в +65535) в 2-байтовое целое число (0- 65535), помещая его в LINNUM.

**GETNUM E746 (59206)**

Считать 2-байтовое число в LINNUM из TXTPTR, проверить наличие запятой и получить однобайтовое число в X. При вводе TXTPTR указывает на первый символ формулы для первого числа, использует подпрограммы FRMNUM GETADR, CHKCOM, GETBYT.

**COMBYTE E74C (59212)**

Проверить наличие запятой и получить байт в X. Использует CHKCOM, GETBYT. При указателе в TXTPTR.

**TXTPTR TO FAC****FRMEVL DD7B (56699)**

Вычислить формулу, на которую указывает TXTPTR, используя CHRGET и оставить результат в FAC. Это основная подпрограмма для команд, которые используют формулы и работают как со строками, так и с числами. Если формула является строковым литералом, FRMEVL пропускает открывающую кавычку и выполняет STRLIT и ST2TXT.

**FRMNUM DD67 (56679)**

Вычислить формулу в TXTPTR, поместить ее в FAC и проверить, что это число. При вводе TXTPTR указывает на первый символ формулы. Выдает ошибку "TYPE MSMATCH ERROR" ("ОШИБКА ТИПА"), если формула является строкой.

**FIN EC4A (60490)**

Ввести число формата с плавающей точкой в FAC из CHRGET. FIN предполагает, что регистры 6502 и A были настроены CHRGET, который извлек первую цифру.

**СТРОКОВЫЕ УТИЛИТЫ**

В Applesoft строки состоят из трех частей: дескриптор, указатель на дескриптор и строка ASCII.

Дескриптор строки содержит длину строки и адрес ее первого символа. См. стр. 137 Справочного руководства Applesoft. В большинстве процедур дескриптор остается в памяти, а указатель хранится в FAC. Указатель — это адрес дескриптора. Фактическая строка может находиться где угодно в памяти. В программе 10 A\$="HI" оставит дескриптор, указывающий на текст начала программы.

**CAT E597 (58775)**

Объединить две строки. FACMO,LO указывает на дескриптор первой строки, а TXTPTR указывает на знак «+».

**STRINI E3D5 (58325)**

Получить место для создания строки в памяти и создать дескриптор для нее в DSCTMP. На входе A=длина строки.

**STRSPA E3DD (58333)**

Обратиться к GETSTA (JSR GETSPA) и сохранить указатель и длину строки в DSCTMP.

**COPY DAB7 (55991)**

Освободить строку, на которую временно указывает Y,A и переместите ее в память, на которую указывает FORPNT.

**MOVINS E5D4 (58836)**

Переместить строку, на дескриптор которой указывает STRNG1, в память, на которую указывает FRESPA.

**MOVSTR E5E2 (58850)**

Переместить строку, на которую указывают Y,X длиной A в память, на которую указывает FRESPA.

**STRTXT DE81 (56961)**

Устанавливает Y,A равным TXTPTR плюс C и переходит к STRUT.

**STRLIT E3E7 (58343)**

Сохраните литерал в ENDCHR и CHARAC, чтобы STRLT2 остановился на ней.

**STRLT2 E3ED (58349)**

Взять строковый литерал, на первый символ которого указывает Y,A, и создать для него дескриптор. Дескриптор строится в DSCTMP, но PUTNEW переводит его во временный и оставляет указатель на него в FACMO,LO. Символы, отличные от нуля, которые завершают строку, должны быть сохранены в CHARAC и ENDCHR. Ведущие кавычки должны быть пропущены перед STRLT2. При выходе на символ после строкового литерала указывает STRNG2. Переходит в PUTNEW.

**PUTNEW E42A (58410)**

Некоторые строковые функции возвращают результат в DSCTMP. Переместить DSCTMP во временный дескриптор, поместить указатель на дескриптор в FACMO,LO и отметить результат как строку.

**GETSPA E452 (58450)**

Получить место для строки символов. Может принудительно запустить сборку мусора. Перемещает FRESPC и FRETOP вниз достаточно для сохранения строки. При входе A = количество символов в строке. Возвращает A неизменным и указатель на место строки в Y,X, FRESPC и FRETOP. Если места в памяти нет, то выдать ошибку "OUT OF MEMORY" (НЕТ ПАМЯТИ).

**FRESTR E5FD (58877)**

Проверить, что последний результат FAC был строкой и перейти в ...



**FREFAC E600 (58880)**

Загрузить указатель дескриптора строки в FACMO,LO в Y и A и перейти к FRETMP.

**FRETMP E604 (58884)**

Освобождение временной строки. При входе указатель на дескриптор находится в Y,A. Проверяется, является ли дескриптор временным, выделенным PUTNEW. Если да, то временный освобождается путем обновления TEMPRT. Если временный освобождается, выполняется дополнительная проверка, чтобы увидеть, является ли строка самой низкой в памяти. Если да, то эта область памяти также освобождается путем обновления FRETOP. При выходе адрес строки находится в INDEX и Y,X, а длина строки в A.

**FRETMS E635 (58933)**

Удалить временный дескриптор строки, не удаляя строку. На входе Y,A указывает на дескриптор, который нужно удалить. На выходе Z устанавливается, если что-то было освобождено.

**ПРОЦЕДУРЫ ВВОДА ДАННЫХ с УСТРОЙСТВА****INLIN D52C (54572) (Без приглашения)****INLIN+2 D52E (54574)**

В регистре X-символ подсказки. Ввести строку текста с текущего устройства ввода во входной буфер BUF и перейти к GDBUFS.

**GDBUFS D539 (54985)**

Помещает ноль в конец входного буфера, BUF, и маскирует старший бит во всех байтах.

На входе: X=конец входной строки (длина ? символ конца строки ?)

На выходе: A=0, X=FF, Y=1

**INCHR D553 (54611)**

Получить один символ из текущего устройства ввода в A и замаскировать старший бит. INCHR использует основные процедуры ввода Apple и поддерживает обычное приветствие.

**ПРОЦЕДУРЫ ВЫВОДА НА УСТРОЙСТВО****STROUT DB3A (56122)**

Печать строки, на которую указывают Y,A. Строка должна заканчиваться нулем или кавычками.

**STRPRT DB3D (56125)**

Печать строки, на дескриптор которой указывает FACMO, FACLO.

**OUTDO DB5C (56156)**

Печать символа в A. Действуют INVERSE, FLASH и NORMAL.

**CRDO DAFB (56059)**

Печать возврата каретки.

**OUTSPC DB57 (56151)**

Печать пробела.

**OUTQST DB5A (56154)**

Печать вопросительного знака.

**INPRT ED19 (60697)**

Печатает "IN" и текущий номер строки из CURLIN. Использует LINPRT.

**LINPRT ED24 (60708)**

Печатает 2-байтовое беззнаковое число в X,A.

**PRNTFAC ED2E (60718)**

Печатает текущее значение FAC. FAC уничтожается. Использует FOUT и STROUT.

## ВНУТРЕННИЕ ПОИСКОВЫЕ ПРОЦЕДУРЫ

### **PTRGET DFE3 (57315)**

Чтение имени переменной из CHRGET и поиск его в памяти. При входе TXTPTR указывает на первый символ имени переменной. При выходе адрес значения переменной находится в VARPNT и Y,A. Если PTRGET не может найти простую переменную, он создает ее. Если он не может найти массив, он создает один размерностью до 10 и устанавливает все элементы равными нулю.

### **GETARYPT F7D9 (63449)**

Чтение имени переменной из CHRGET и поиск его в памяти. При входе TXTPTR указывает на первый символ имени переменной. Эта процедура оставляет LOWTR, указывающим на имя массива переменных. Если массив не найден, результатом будет ошибка "OUT OF DATA ERROR" (НЕТ ДАННЫХ).

### **FNDLIN D61A (54810)**

Выполняет поиск в программе строки, номер которой находится в LINNUM.

При выходе:

1. Если C установлен, LOWTR указывает на поле ссылки нужной строки.
2. Если C очищен, то строка не найдена. LOWTR на следующую более высокую строку.

### **DATA D995 (55701)**

Переместить TXTPTR в конец оператора. Ищет ':' или eol (0).

### **DATAN D9A3 (55715)**

Рассчитайте смещение по Y от TXTPTR до следующего «:» или eol (0).

### **REMN D9A6 (55718)**

Рассчитать смещение по Y от TXTPTR до следующего конца строки (0).

### **ADDON D998 (55704)**

Добавить Y TXTPTR.

## ПРОЦЕДУРЫ ИНИЦИАЛИЗАЦИИ

**SCRTCH D64B (54859)** Команда "NEW". Очищает программу, переменные и стек.

**CLEARC D66C (54892)** Команда "CLEAR". Очищает переменные и стек.

**STKINI D683 (54915)** Очищает стек.

**RESTOR D849 (55369)** Устанавливает указатель DATA, DATPTR, в начало программы.

**STXTPT D697 (54935)** Устанавливает TXTPTR в начало программы.

## ПРОЦЕДУРЫ УПРАВЛЕНИЯ ХРАНИЕНИЕМ

### **BLTU D393 (54163)**

Передача блока (страницы ?) освобождает место, перемещая все вперед.

На входе: Y,A и HIGHDS=назначение высокого адреса +1; LOWTR=самый низкий адрес для перемещения; HIGHTR=самый высокий адрес для перемещения + 1

На выходе: LOWTR не изменяется; HIGHTR=LOWTR - \$100; HIGHDS=самый низкий адрес для перемещения - \$100

### **REASON D3E3 (54243)**

Убеждается, что в памяти достаточно места, проверяет, что адрес Y,A меньше FRETOP.

Может вызвать сборку мусора. Вызывает OMERR (?), если места нет.

### **GARBAG E484 (58500)**

Сборка мусора. Переместить все используемые в данный момент строки в памяти как можно дальше. Это максимизирует свободную область памяти для большего количества строк или числовых переменных.

## РАЗНОЕ. ОСНОВНЫЕ КОМАНДЫ

Обратите внимание, что многие команды не документированы, поскольку они переходят в новый выборщик операторов и не могут использоваться как подпрограмма.

**CONT D898 (55448)**

Перемещает OLDTXT и OLDLIN в TXTPTR и CURLIN. Начало выполнения программы пользователя.

**NEWSTT D7D2 (55250)**

Выполнить новый оператор. При вводе TXTPTR указывает на '.', предшествующий оператору, или на ноль в конце предыдущей строки. Используйте NEWSTT для перезапуска программы с CONT. ! Эта подпрограмма не возвращается (нет RETURN)

**RUN D566 (54630)**

Запуск Бейсик-программы. ! Нет возврата (нет RETURN)

**GOTO D93E (55614)**

Использует LINGET и FNDLIN для обновления TXTPTR. GOTO предполагает, что регистры 6502 и A были установлены CHRGET, который извлек первую цифру номера перехода к строке

**LET DA46 (55878)**

Использует CHRGET для получения адреса переменной, '=', вычисления формулы и сохранения ее. При вводе TXTPTR указывает на первый символ имени переменной.

## ПОДПРОГРАММЫ ГРАФИКИ ВЫСОКОГО РАЗРЕШЕНИЯ

ПРИМЕЧАНИЕ: Независимо от того, какой экран отображается, HPAG (расположение \$E6) определяет, на каком экране происходит рисование. (\$20 для HGR, \$40 для HGR2)

**HGR2 F3D8 (62424)**

Инициализация и очистка страницы 2 HIRES.

**HGR F3E2 (62434)**

Инициализация и очистка страницы 1 HIRES.

**HCLR F3F2 (62450)**

Очистка экрана HIRES до черного цвета.

**BKGND F3F6 (62454)**

Очистка экрана HIRES до последнего цвета, нанесенного на график.

**HPOSN F411 (62481)**

Позиционирует курсор HIRES без построения графика, HPAG определяет, на какую страницу направлен курсор. При входе: Горизонтально = Y, X ; Вертикально = A.

**HPLT F457 (62551)**

Вызывает HPOSN, затем пытается нанести на график точку в позиции курсора. Точка не может быть нанесена, если нанесен не белый цвет на координату X дополнительного цвета.

При входе: Горизонтально=Y,X; Вертикально=A.

**HLIN F53A (62778)**

Рисует линию от последней нанесенной на карту точки или конечной точки линии до координаты в регистрах 6502.

При входе: Горизонтально=X,A; Вертикально=Y.

**HFIND F5CB (62923)**

Преобразует положение курсора HIRES в координаты X-Y. Используется после SHAPE для поиска того, где вы остановились. При выходе: \$E0=горизонтальный lsb; \$E1=горизонтальный msb; \$E2=вертикальный.

**DRAW F601 (62977)**

Рисует форму, на которую указывают Y,X используя текущий HCOLOR. При входе A=коэффициент вращения.

**XDRAW F65D (63069)**

Рисует фигуру, на которую указывает Y,X, инвертировав существующий цвет точек, которые составляют фигуру. При входе A= коэффициент вращения.

**SETHCOL F6EC (63212)** Установить цвет HIRES их регистра X. X должен быть меньше 8.

**SHLOAD F775 (63349)**

Загружает таблицу фигур в память с ленты выше MEMSIZ (HIMEM) и устанавливает указатель на \$E8.

**ПОДПРОГРАММЫ МАГНИТОФОННОЙ ЛЕНТЫ**

**SAVE D8B0 (55472)** Сохранить программу из памяти на ленте.

**LOAD D8C9 (55497)** Загрузить программу с ленты.

**VARTIO D8F0 (55536)** Настроить A1 и A2, чтобы сохранить 3 байта (\$50-\$52) для длины.

**PROGIO D901 (55553)** Настроить A1 и A2, чтобы сохранить текст программы.

**ПОДПРОГРАММЫ ОБРАБОТКИ ОШИБОК****ERROR D412 (54290)**

Проверяет ERRFLG и переходит к HNDLERR, если ONERR активен. В противном случае он выводит [c/r] '?' [сообщение об ошибке § X] 'ERROR'. Если это происходит во время выполнения программы, то он также выводит 'IN' и CURLIN.

**HANDLERR F2E9 (62185)**

Сохраняет CURLIN в ERRLIN, TXTPTR в ERRPOS, X в ERRNUM и REMSTK в ERRSTK. REMSTK равняется указателю стека 6502 и устанавливается в начале каждого оператора. X содержит код ошибки. Это может использоваться для прерывания выполнения программы BASIC. См. Applesoft Reference Manual на стр. 136 для значения X для данной ошибки.

**RESUME F317 (62231)**

Восстанавливает CURLIN из ERRLIN и TXTPTR из ERRPOS и переносит ERRSTK в указатель стека 6502.

**ПРОЦЕДУРЫ ПРОВЕРКИ СИНТАКСИСА****ISCNTC D858 (55384)**

Проверяет клавиатуру Apple на наличие Упр — С (\$83). Выполняет процедуру BREAK если нажата УПР — С.

**CHKNUM DD6A (56682)** Проверить, что в FAC содержится число. См. CHKVAL.

**CHKSTR DD6C** Проверить, что FAC является строкой. См. CHKVAL.

**CHKVAL DD6D (56685)**

Проверяет результат последней операции FAC, чтобы узнать, является ли он строкой или числовой переменной. Если FAC и C не совпадают, возникает ошибка "TYPE MISMATCH ERROR" (ОШИБКА ТИПА).

При входе: C=1 проверяет наличие строк; C=0 проверяет наличие числовых данных

**ERRDIR E306 (58118)**

Вызывает "ILLEGAL DIRECT ERROR" (НЕ В ДИАЛОГЕ), если программа не запущена. X изменен.

**ISLETC E07D (57469)**

Проверяет A на наличие буквы ASCII (от 'A' до 'Z'). На выходе C устанавливает, является ли A буквой.

**PARCHK DEB2 (57010)**

Проверяет символ строки на совпадение с открывающей скобкой '(', вычисляет формулу и проверяет наличие закрывающей скобки ')'. Использует CHKOPN и FRMEVL, затем переходит к CHKCLS.

**CHKCLS DEBS (57016)** Проверяет в TXTPTR на наличие ')'. Использует SYNCHR.

**CHKOPN DEBB (57019)** Проверяет TXTPTR на наличие '(', Использует SYNCHR.

**CHKCOM DEBE (57022)** Проверяет TXTPTR на наличие ','. Использует SYNCHR.

**SYNCHR DEC0 (57024)**

Проверяет TXTPTR на наличие символа в A. TXTPTR не изменяется. Обычно выходит через CHRGET. Выходит с "SYNTAX ERROR" (СИНТАКСИЧЕСКАЯ ОШИБКА), если они не совпадают.

All About Applesoft 56

"DTCA2DOC-043-06.PICT" 260 KB 2001 -04-03 dpi: 300h x 300v pix: 2244h x 301 8v

Source: David T Craig Page 0007 of 0008

Apple 2 Computer Information • Document 043

# Указатель

— A —		— F —		— I —		— Q —	
A1 3C,3D	51	FADO E7BE	52	INDEX 5E,5F	51	QINT EBF2	53
A2 3E,3F	51	FADDH E7A0	53	INCUR D553	54	— R —	
ABS EBAF	53	FADDT E7C1	52	INLIN D52C	54	REASON D3E3	55
ADDON D998	55	FBUFFR 100-1FF	51	INLIN+2 D52E	54	REMN D9A6	55
ARYTAB 6B,6C	51	FCOMP EBB2	53	INPRT ED19	54	REMSTK F8	51
ATN F09E	53	FDIV EA66	52	INT ED23	53	RESTOR D849	55
AYINT EIOC	53	FDIVT EA69	52	INVFLG 32	51	RESUME F317	56
— B —		FIN EC4A	53	ISCNTC D858	56	RND EFAE	53
BKGND F3F2	55	FIRST FO	51	ISLETC E07D	56	ROT F9	51
BLTU D393	55	FLOAT EB93	53	— L —		RUN D566	55
BUF 200-2FF	51	FMULT E97F	52	LASTPT 53	51	— S —	
— C —		FMULTT E982	52	LET DA46	55	SAVE D8B0	56
CAT E597	54	FNDLIN D61A	55	LINGET DAOC	52	SCALE E7	51
CHARAC 0D	51	FORPNT 85,86	51	LINNUM 50,51	51	SCRTCH D64B	55
CHKCLS DEBS	56	FOUT ED34	53	LINPRT ED24	54	SETHCOL F6EC	56
CHKCOM DEBE	56	FPWRT EE97	52	LOAD D8C9	56	SGN EB80	53
CHKNUM DD6A	56	FREFAC E600	54	LOG E941	53	SHLOAD F775	56
CHKOPN DEBB	56	FRESPEC 71,72	51	LOWTR 9B,9C	51	SIGN EB82	53
CHKSTR DD6C	56	FRESTR E5FD	54	— M —		SIN EFFI	53
CHKVAL DD6D	56	FRETMP E604	54	MEMSIZ 73,74	51	SNGFLT E301	53
CHRGET OOB1	52	FRETMS E635	54	MOVIF EB21	53	SPDBYT FI	51
CHRGOT 00B7	52	FRETOP 6F,70	51	M0V2F EB1E	53	SQR EE8D	53
CLEARC D66C	55	FRMEVL DD7B	53	MOVAF EB63	53	STKINI 0683	55
COMBYTE E74C	53	FRMNUM DD67	54	MOVFA EB53	53	STREND 6D,6E	51
CONINT E6FB	53	FSUB E7A7	52	MOVFM EAF9	53	STRINI E3D5	54
CONT D898	55	FSUBT E7AA	52	MOVINS E5D4	54	STRLT E3E7	54
CONUPK E9E3	53	— G —		MOVML EB23	53	STRLT2 E3ED	54
COPY DAB7	54	GARBAG E484	55	MOVSTR E5E2	54	STRNG1 AB,AC	51
COS EFEA	53	GDBUFS D539	54	MULIO EA39	53	STRNG2 AD,AE	51
CRDO DAFB	54	GETADR E752	53	— N —		STROUT DB3A	54
CURLIN 75,76	51	GETARYPT F7D9	55	NEGOP EEDO	53	STRPRT DB3D	54
— D —		GTBYTC E6F5	52	NEWSTT D7D2	55	STRSPA E3D0	54
DATA D995	55	GTBYT E6F8	52	— O —		STRTXT DE81	54
DATAN 09A3	55	GETNUM E746	53	OLDLIN 77,78	51	STXTPT D697	55
DATLIN 7B,7C	51	GETSPA E452	54	ORMASK F3	51	SUBFLG 14	51
DATPTR 7D,7E	51	GIVAYF E2F2	53	OUTDO DB5C	54	SYNCHR DEC0	56
DIVIO EA55	53	GOTO D93E	55	OUTQST DB5A	54	— T —	
DRAW F601	55	— H —		OUTSPC DB57	54	TAN F03A	53
OSCTMP 9D-9F	51	H2 2C	51	— P —		TEMPPT 52	51
— E —		HANOLERR F2E9	56	PARCHK DEB2	56	TXTPTR B8,B9	51
ENDCHR OE	51	HCLR F3EE	55	PLOTFNS FIEC	52	TXTTAB 67,68	51
ERRDIR E306	56	HEND F5CB	55	PRGENO AF,BO	51	— V —	
ERRFLG D8	51	HFNS F6B9	52	PROG 10 D901	56	V2 20	51
ERRLIN DA,DB	51	HGR F3DE	55	PRNTFAC ED2E	54	VALTYP 11	51
ERRNUM DE	51	HGR2 F3D4	55	PTRGET DFE3	55	VARPNT 83,84	51
ERROR D412	56	HIGHDS 94,95	51	PUTNEW E42A	54	VARTAB 69,6A	51
ERRPOS DC,DD	51	HIGHTR 96,97	51	— X —		VARTIO D8F0	56
ERRSTK DF	51	HLIN F530	55	XDRAW F650			
EXP ER09	53	HPAG E6	51				
		HPLLOT F453	55				
		HPOSN F40D	55				