

# MSX Network BASIC

Используется в решении [Yamaha Локальная сеть, версия 3.0](#).

## Общие сведения о локальной сети

Назначение сети.

Данная локальная сеть разработана для связи компьютеров в учебном классе. Она позволяет соединять преподавателя и 15 учеников, давая при этом возможность обмена программами, данными и сообщениями, как между преподавателем и учеником, так и между двумя учениками. Сеть работает не только в [MSX BASIC](#), но и в [MSX Disk BASIC](#), [MSX-DOS](#) и [MSX CP/M](#).

Примечания:

1. Ниже, для простоты записи, вместо «компьютер преподавателя» и «компьютер ученика» будет употребляться «преподаватель» и «ученик».
2. Как и в предыдущих версиях сети, в версии 3.0 при обращении ученика к дисководу, команды сети [MSX BASIC](#) временно отключаются.
3. Для работы сети используется специальное ОЗУ (2 кбайт) в сетевом блоке, обычное ОЗУ при работе с сетью не используется. Для обращения к сетевому ОЗУ, используйте карту памяти.
4. Вызов сетевой BIOS (системы базовых функций ввода-вывода) из [MSX-DOS](#) или [MSX CP/M](#) ссылается на рабочую область из 8 байт в основном ОЗУ.
5. Если при работающей сети невозможно запустить некоторые прикладные программы (например [Графический редактор Yamaha Painter](#)), воспользуйтесь командой [CALL NETEND](#), а после неё повторите вызов Вашей программы.

## Список дополнительных операторов MSX BASIC

Команда	Описание	Кем может быть вызвана	
		преподавателем	учеником
<a href="#">CALL HELP</a>	выводит этот список с форматами	✓	✓
<a href="#">CALL WHO</a>	возвращает номер Вашего компьютера	✓	✓
<a href="#">CALL SNDRUN</a>	передача и запуск программы на <a href="#">MSX BASIC</a>	✓	—
<a href="#">CALL SEND</a>	передача программы на <a href="#">MSX BASIC</a>	✓	—
<a href="#">CALL BSEND</a>	передача программы в машинном коде или изображения с экрана	✓	✓
<a href="#">CALL RECEIVE</a>	приём программы на <a href="#">MSX BASIC</a>	✓	✓
<a href="#">CALL BRECEIVE</a>	приём программы в машинных кодах или изображения с экрана	✓	—
<a href="#">CALL SNDMAIL</a>	передача информации в почтовый ящик преподавателя	✓	—
<a href="#">CALL RCVMAIL</a>	приём информации из почтового ящика преподавателя	✓	—
<a href="#">CALL MESSAGE</a>	передача сообщения преподавателем	✓	—
<a href="#">CALL SNDCMD</a>	передача команды <a href="#">MSX BASIC</a>	✓	—
<a href="#">CALL RUN</a>	запуск программы на <a href="#">MSX BASIC</a> у ученика	✓	—
<a href="#">CALL STOP</a>	остановка программы на <a href="#">MSX BASIC</a> у ученика	✓	—
<a href="#">CALL POKE</a>	запись числа в память к ученику или в сетевое ОЗУ	✓	✓
<a href="#">CALL PEEK</a>	чтение числа из памяти у ученика и из сетевого ОЗУ	✓	✓
<a href="#">CALL PON</a>	начало упорядоченного опроса	✓	—
<a href="#">CALL POFF</a>	конец упорядоченного опроса	✓	—
<a href="#">CALL CHECK</a>	проверка кто из учеников подключён к сети	✓	—

Команда	Описание	Кем может быть вызвана	
		преподавателем	учеником
CALL TALK	передача сообщения учеником	—	✓
CALL ONLINE	включение в сеть (только после команды CALL NETINIT)	—	✓
CALL OFFLINE	отключение от сети (только после команды CALL NETINIT)	—	✓
CALL NETEND	выключение сети	✓	✓
CALL NETINIT	инициализация сети	✓	✓
CALL ENACOM	разрешение передачи учеником	✓	—
CALL DISCOM	запрещение передачи учеником	✓	—

Все команды серии **CALL** можно сократить до 3–4 букв, а само слово **CALL** можно заменить символом «\_» (подчёркивание) после которого не требуется пробелов, например, **CALL NETEND** можно заменить на **\_NETE**

## CALL HELP

Команда преподавателя/ученика

Формат:

```
CALL HELP
```

Показывает список команд с форматами.

Эта команда показывает список сетевых команд **MSX BASIC**, которые используются при работе с сетью в классе. Экран должен быть в текстовом режиме. Так как наборы команд преподавателя и ученика различны, списки у них будут соответственно отличаться. У ученика, кроме того, указывается номер его компьютера. Эта функция у ученика работает вне зависимости от команды разрешения передачи (**CALL ENACOM**). Когда передача разрешена, в список добавляются команды: **CALL SEND**, **CALL RECEIVE**, **CALL BSEND**, **CALL BRESEIVE** и **CALL TALK**.

## CALL WHO

Команда преподавателя/ученика

Формат:

```
CALL WHO [(<переменная>)]
```

Возвращает номер Вашего компьютера.

*Пример 1*

```
CALL WHO (A)
PRINT A
3
Ok
```

Эта команда возвращает номер компьютера, установленный при помощи переключателя в сетевом блоке (для преподавателя — 0, для учеников — 1–15). В этом примере, команда возвращает номер 3 ученика.

*Пример 2*

```
CALL WHO
3
```

Когда переменная отсутствует, номер сразу выводится на экран.

Для использования этого формата Вы должны быть в командном режиме.

## CALL BSEND

Команда преподавателя/ученика

Формат:

```
CALL BSEND
```

Формат:

```
CALL BSEND ([<имя файла>],[<номер ученика>],[<начальный адрес>],[<конечный адрес>],[<S>])
```

Передача программ в машинном коде или изображения с экрана.

### Пример 1

```
CALL BSEND ("DATA.BIN",12,&H8000)
```

Эта команда переадресует программу в машинных кодах или данные, записанные из **MSX BASIC** командой **BSAVE** или **CALL BRECEIVE**, по начальному адресу в память ученика, определённого номером. В этом примере файл «DATA.BIN» пересылается в память ученика номер 12 с начального адреса &H8000. Если номер ученика не указан начальный адрес, программа или данные пересылаются с того адреса, с которого они были записаны.

### Пример 2

```
CALL BSEND (,12,&H0100,&H2000,S)
```

Когда Вы указываете <S>, содержимое Вашего экрана (видеопамяти) копируется в видеопамять ученика с указанным номером. Перед передачей режим видеопроцессора в принимающем компьютере автоматически меняется на тот режим который находится передающий. В этом примере содержимое видеопамяти преподавателя с адреса &H0100 до адреса &H2000 передаётся по адресу с &H0100 по &H2000 на экран (в видеопамять) ученика номер 12. Когда номер ученика отсутствует, информация передаётся всем ученикам.

```
CALL BSEND (,12,,,S)
```

Когда используется <S> и адреса не указаны, то вместо начального берётся &H0000, а вместо конечного &HFFFF.

### Пример 3

```
CALL BSEND ("DATA.BIN",12,&H0100,&H2000,S)
```

Эта команда передаёт данные с экрана (из видеопамяти), записанные с помощью команд **MSX BASIC BSAVE** или **CALL BRECEIVE** с использованием <S>, в область с начального по конечный адреса в видеопамять указанного ученика. Когда данные в видеопамять пересылаются из файла, режим видеопроцессора автоматически не устанавливается, т.е. он должен быть установлен до использования команды **CALL BSEND**. В этом примере данные из файла «DATA.BIN» передаются по адресам с &H0100 по &H2000 в видеопамять ученика с номером 12. Если номер ученика не указан данные передаются всем ученикам. Если не указан начальный адрес данные будут пересылаться начиная с того адреса, с которого они были записаны. Если этот адрес больше чем конечный, то используется именно он, а конечный игнорируется. А если меньше, то данные будут посланы лишь до конечного адреса, указанного в команде **CALL BSEND**.

## CALL BRECEIVE

Команда преподавателя

Формат:

```
CALL BRECEIVE ([<имя файла>],[<номер ученика>],[<начальный адрес>],[<конечный адрес>] , [<S>])
```

Приём программы в машинных кодах или изображения с экрана

#### Пример 1

```
CALL BREC ("DATA.BIN" , 10 , &HA000 , &HB000)
```

Эта команда записывает программу в машинных кодах данные из памяти, с начального адреса по конечный, в файл на диск. Формат для записи такой же, как и в команде [MSX BASIC BSAVE](#), только стартовым адресом всегда является начальный. В этом примере содержимое памяти ученика номер 10, с адреса &HA000 по &HB000, записывается в файл «DATA.BIN».

#### Пример 2

```
HCALL BREC ( , 10 , &H0100 , &H2000 , S)
```

Когда используется <S>, содержимое экрана (видеопамяти) ученика с указанным номером принимается в видеопамять преподавателя. Перед этим видеопроцессор автоматически устанавливается в соответствующий режим. В этом примере содержимое видеопамяти ученика номер 10 с адреса &H0100 по &H2000 пересылаются по этим же адресам на экран (в видеопамять) преподавателя. При использовании <S>, если Вы не указываете начальный адрес, используется — &H0000, а вместо конечного — &HFFFF.

#### Пример 3

```
CALL BREC ("DATA.BIN" , 12 , &H0100 , &H2000 , S)
```

Содержимое экрана (видеопамяти) ученика принимается и записывается на диск. Формат для записи такой же, как и в команде [MSX BASIC BSAVE](#) при использовании <S>. Так как записываемый файл (содержимое видеопамяти) не содержит информацию о режиме видеопроцессора, необходимо при использовании такого файла предварительно установить соответствующий режим. В этом примере данные с экрана (из видеопамяти) ученика номер 12 с адреса &H0100 по &H2000 записывается на диск, в файле с названием «DATA.BIN». При использовании <S>, если Вы не указываете начальный адрес, используется — &H0000, а вместо конечного — &HFFFF.

## CALL ENACOM

Команда преподавателя

Формат:

```
CALL ENASOM (<номер ученика>)
```

Разрешает передачу учеником.

#### Пример

```
CALL ENAC (12)
```

Эта команда разрешает указанному преподавателем ученику передачу в сеть. Этот пример разрешает связь ученику номер 12, т.е. он может использовать команды [CALL SEND](#), [CALL RESEIVE](#), [CALL BSEND](#), [CALL BRESEIVE](#). При указанном 0 передача в сеть разрешается всем ученикам. Ученик, которому передача запрещена, может посылать сообщения другим ученикам командой [CALL TALK](#).

## CALL DISCOM

Команда преподавателя

Формат:

```
CALL DISCOM (<номер ученика>)
```

Запрещает передачу учеником.

*Пример*

```
HCALL DISC (12)
```

Эта команда запрещает указанному преподавателем ученику передачу в сеть. Этот пример запрещает связь ученику номер 12, т.е. он не может использовать команды [CALL SEND](#), [CALL RESEIVE](#), [CALL BSEND](#), [CALL BRESEIVE](#). При указании 0 передача в сеть запрещена всем ученикам. При начале работы с сетью, передача в сеть запрещена всем ученикам.

## CALL CHECK

Команда преподавателя

Формат:

```
CALL CHECK ([<переменная>],[<переменная>])
```

Проверяет кто из учеников подключён к сети и кому из учеников разрешена передача в сеть.

*Пример 1*

```
CALL CHECK (A)
PRINT BIN$ (A)
110110111101100
```

Эта команда проверяет, кто из учеников подключён к сети и возвращает двоичное число, в котором младший разряд соответствует ученику номер 1, следующий номер 2 и т.д. Когда соответствующий бит равен 0, это означает, что ученик подключён к сети, а когда равен 1 — отключён от неё. В этом примере к сети подключены ученики номер 1, 2, 5, 10 и 13. Когда все ученики подключены, значение возвращаемой переменной равно 0. То, что ученик отключён от сети означает: либо, что компьютер выключен или сеть физически от него отключена, либо, что это сделано программно при помощи команды [CALL OFFLINE](#).

*Пример 2*

```
HCALL CHECK (,B)
PRINT BIN$ (B)
111111111101110
```

Эта команда проверяет, кому из учеников разрешена передача в сеть, и возвращает двоичное число, в котором младший разряд соответствует ученику номер 1, следующий номер 2 и т.д. Когда ответствующий бит равен 0, это означает, что передача в сеть этому ученику разрешена а когда равен 1 — запрещена. В том примере передача разрешена ученикам номер 1 и 5. Когда передача разрешена всем ученикам значение возвращаемой переменной равна 0. Разрешение и запрещение передачи учеником в сеть производится соответственно командами [CALL ENACOM](#) и [CALL DISCOM](#).

## CALL NETINIT

Команда преподавателя

Формат:

```
CALL NETINIT
```

Инициализация сети.

Пример

```
CALL NETINIT
```

Эта команда используется для начала работы с сетью, если та не была инициализирована при включении компьютеров. В этом случае без подачи преподавателем этой команды сеть работать не будет. Обычно сеть инициализируется при включении.

## CALL NETEND

Команда преподавателя/ученика

Формат:

```
CALL NETEND
```

Отключение сети.

Пример

```
CALL NETEND
```

Эта команда используется для отключения сети, когда Вы хотите работать с прикладной программой, которая не может быть вызвана при работающей сети.

## CALL POKE

Команда преподавателя/ученика

Формат:

```
CALL POKE (<число>,<адрес>,[<номер ученика>] , [<N>])
```

Запись числа в память ученика или в сетевое ОЗУ.

Пример 1

для преподавателя/ученика

```
CALL POKE (100,&H7800)
```

Эта команда записывает указанное число по указанному адресу в сетевое ОЗУ (&H7800-&H7FFF). В этом примере 100 записывается по адресу сетевых сообщений &H7800.

Пример 2

для преподавателя

```
CALL POKE (100,&HB000,1)
```

Эта команда записывает данное число по указанному адресу память указанному ученику. Если в качестве номера указан 0, то число записывается в память всем ученикам. В этом примере 100 записывается по адресу &HB000 в память ученику номер 1.

Пример 3

для преподавателя

```
CALL POKE (100,&H7A00,1,N)
```

Эта команда записывает данное число по указанному адресу в сетевое ОЗУ (&H7800-&H7FFF) означенному ученику. В этом примере 100 записывает по адресу &H7A00 в сетевое ОЗУ ученику номер 1.

## CALL PEEK

Команда преподавателя/ученика

Формат:

```
CALL PEEK (<переменная>,<адрес>,<номер ученика> , [<N>])
```

Читает число из памяти ученика или из сетевого ОЗУ.

### Пример 1

для преподавателя/ученика

```
CALL PEEK (A,&H7800)
```

Эта команда читает ячейку по данному адресу в сетевом ОЗУ (&H7800-&H7FFF) и возвращает в указанной переменной. В этом примере читаются данные из ячейки сетевых сообщений с адреса &H7800 и возвращаются в переменную A.

### Пример 2

для преподавателя

```
CALL PEEK (A,&HB00,1)
```

Эта команда читает данные по данному адресу из памяти указанного ученика и возвращает значение в заданной переменной. Если в качестве номера 0, то генерируется ошибка. В этом примере читаются данные из ячейки памяти с адресом &HB000 ученика номер 1 и возвращаются в переменную A.

### Пример 3

для преподавателя

```
CALL PEEK (A,&H7A00,1,N)
```

Эта команда читает данные по данному адресу в сетевом ОЗУ (&H7800-&H7FFF) указанного ученика и возвращает значение в заданной переменной. В этом примере читается данные из ячейки сетевого ОЗУ с адресом &HB000 ученик номер 1 и возвращается в переменной A.

## CALL MESSAGE

Команда преподавателя

Формат:

```
CALL MESSAGE (<сообщение>,<номер ученика>)]
```

Передача сообщения преподавателем.

### Пример

```
CALL MESSAGE ("Hello !!",10)
```

Эта команда передаёт сообщение преподавателя одному или нескольким ученикам. Сообщение имеет максимальную длину 56 символов, но, когда оно высвечивается в 24 строке на экране ученика, его длина зависит от того, в каком режиме сейчас ученик. Если в момент передачи ученик находится в графическом режиме, то сообщение будет выведено к нему на экран, как только он выйдет в текстовый режим. Если параметр <номер ученика> пропущен,

сообщение передаётся всем ученикам. Этот пример высвечивает сообщение «Hello !!» в 24 строке на экран у 10 ученика. Чтобы сообщение исчезло — нажмите пробел.

Примечание:

Если у ученика выключена индикация функциональных клавиш (по команде [KEY OFF](#)), то сообщение будет двигаться вверх по мере заполнения экрана, т.е. будет рассматриваться как обычный текст.

## CALL TALK

Команда ученика

Формат:

```
CALL TALK <сообщение>,[<переменная>]
```

Передача сообщения учеником.

*Пример*

```
CALL TALK ("Не могу понять",B)
```

Когда ученику не разрешена передача, он может послать сообщение только преподавателю, а когда она разрешена то и другим ученикам. Пользуясь этой командой, ученик должен определить номер ученика/преподавателя, используя переменную. Если значение переменной равно 0, сообщение передаётся преподавателю. Если от 1 до 15 — соответствующему ученику. Если передача прошла успешно, в переменной возвращается 0, если нет — 255. Сообщение имеет максимальную длину 56 символов, но, когда оно высвечивается в 24 строке на экране, его длина зависит от того, в каком режиме сейчас компьютер. В этом примере сообщение «Не могу понять» передаётся преподавателю/ученику, определённой в переменной B.

## CALL SEND

Команды преподавателя

Формат:

```
CALL SEND [( [имя файла],[номер ученика]])]
```

Пересылает ученику программу на [MSX BASIC](#).

*Пример 1*

```
CALL SEND ("A:TEST.BAS",0)
```

Эта команда считывает заданную программу на Бейкике с диска и посылает её указанному ученику. Если ученик работает с программой на [MSX BASIC](#), его программа будет стёрта и он получит новую. Во время передачи у него на экране будет высвечено сообщение «Wait» (ждите), а как только пересылка будет закончена — сообщение «Ок».

Если номер ученика опускается или равен 0 — программа передаётся всем ученикам. Эта команда не стирает содержимое памяти преподавателя и может быть использована в программном режиме. В этом примере программа на [MSX BASIC](#) «TEST.BAS» пересылается всем ученикам.

*Пример 2*

```
CALL SEND (,10)
```

Эта команда передаёт программу, находящуюся в памяти преподавателя. В этом примере программа преподавателя пересылается ученику номер 10. Эта команда может быть использована в программном режиме. Во время передачи, на экране ученика будет высвечено «Wait» (ждите). Если номер ученика опускается, программа передаётся всем ученикам.

# CALL SNDRUN

Команда преподавателя

Формат:

```
CALL SNDRUN ([<имя файла>],[<номер ученика>]])
```

Пересылает ученику программу на [MSX BASIC](#) и запускает её.

*Пример 1*

```
CALL SNDRUN ("A:TEST.BAS",0)
```

Эта команда считывает данную программу на [MSX BASIC](#) с диска, посылает её указанному ученику и запускает её. Если ученик работает с программой на [MSX BASIC](#), его программа будет стёрта и он получит новую, которая и будет запущена. Во время передачи, на экране ученика будет высвечено сообщение «Wait» (ждите). Как только программа будет получена она сразу запускается. Если номер ученика опускается или равен 0 — программа передаётся всем ученикам. Эта команда не стирает содержимое памяти преподавателя и может быть использована в программном режиме. В этом примере программа на [MSX BASIC](#) «TEST.BAS» пересылается и запускается у всех учеников.

Пример 2

```
CALL SNDRUN (,10)
```

Эта команда передаёт программу, находящуюся в памяти преподавателя и запускает её у ученика. В этом примере программа преподавателя пересылается ученику номер 10 и запускается. Эта команда может быть использована в программном режиме. Во время передачи на экране ученика будет высвечено сообщение «Wait» (ждите). Если номер ученика опускается, программа передаётся и запускается у всех учеников.

# CALL RECEIVE

Команда преподавателя

Формат:

```
CALL RECEIVE ([<имя файла>],<номер ученика>)
```

Принимает от ученика программу на [MSX BASIC](#).

*Пример 1*

```
CALL RECE ("B:TEST.BAS",14)
```

Эта команда принимает программу на [MSX BASIC](#) от указанного ученика и записывает её на диск в заданный файл. Команда не стирает содержимое памяти преподавателя и может быть использована в программном режиме. Формат, в котором записывается файл такой же, как и в команде [SAVE MSX BASIC](#).

В этом примере программа ученика номер 14 записывается в файл «TEST.BAS». Если файл с таким именем уже есть на диске, старый будет стёрт и в замен него будет записана программа ученика. Если программа ученика в момент подачи команды работала, она будет остановлена и высветится «Wait» (ждите), а после окончания приёма её преподавателем — «Ok».

*Пример 2*

```
CALL RECE (,1)
```

Эта команда пересылает программу ученика в память преподавателя, стирая при этом программу находящуюся в памяти. Если команда используется в командном режиме, то старая программа прерывается в строке с этой

командой, стирается, а взамен принимается программа ученика, после чего, высвечивается «Ok».

Во время приёма на экране ученика высвечивается сообщение «Wait», а после чего — «Ok». В этом примере программа ученика номер 1 пересылается в текстовом области памяти преподавателя.

## CALL SNDMAIL

Команда преподавателя

Формат:

```
CALL SNDMAIL [( < номер ученика >)]
```

Передаёт содержимое почтового ящика.

Почтовый ящик — это область памяти для передаваемой информации, которая резервируется как в памяти ученика, так и преподавателя. В каждом компьютере есть почтовый ящик для передачи и приёма. Под каждую из этих областей выделяется по 256 байт. Адреса почтовых ящиков записаны в рабочей области (смотрите соответствующий раздел).

Передача заключается в пересылке содержимого из передающего ящика преподавателя в приёмный ящик ученика . Почтовый ящик, таким образом — область для передачи данных. Как Вы её будете использовать — полностью зависит от Вас.

### Пример 1

```
10 FOR L=1 TO 5  
20 CALL SNDM (L)  
30 NEXT L
```

Эта команда копирует содержимое передающего ящика преподавателя в принимающие ящики учеников. В этом примере копируется ученикам 1-5.

### Пример 2

```
CALL SNDM
```

Когда номер ученика пропущен, содержимое ящика преподавателя пересылается в ящики всех учеников, как в этом примере.

## CALL RCVMAIL

Команда преподавателя

Формат:

```
CALL RCVMAIL (<номер ученика>)
```

Принимает содержимого почтового ящика.

### Пример

```
CALL RCVM (5)
```

Эта команда принимает содержимое передающего ящика ученика и записывает его в приёмный ящик преподавателя. В этом примере у пятого ученика.

## CALL SNDCMD

Команда преподавателя

Формат:

```
CALL SNDCMD (<команда>,[<номер ученика>])
```

Передаёт команду на [MSX BASIC](#).

*Пример 1*

```
CALL SNDC ("KEY OFF",3)
```

Эта команда передаёт указанному ученику команду на [MSX BASIC](#) и выполняет её. В конце команды всегда добавляется код перевода каретки (CR). Если номер ученика опущен, команда передаётся всем ученикам. В этом примере в результате выполнения команды, у ученика номер 3 перестают светиться функциональные клавиши.

Если команда набрана с ошибкой или это вообще не команда Бейсика, она все равно будет передана в компьютер ученика и он попытается её выполнить. В этом случае ученик получит сообщение об ошибке. Будьте внимательны, т.к. на компьютере преподавателя никакого сообщения не будет. Если во время передачи команды, у ученика работает программа, она будет остановлена, после чего передаваемая команда будет принята и выполнена.

*Пример 2*

```
10 A$="COLOR 15,4,7"  
20 B$=CHR$(13)  
30 C$="CLS"  
40 CALL SNDC (A$+B$+C$)
```

Можно посылать несколько команд одновременно. В этом примере у всех учеников выполняется команда `color 15,4,7`, а затем — команда `CLS`. При этом между отдельными командами надо вставлять код перевода каретки (CR).

## CALL RUN

Команда преподавателя

Формат:

```
CALL RUN [ ( [ <номер ученика> ], [ <номер строки> ] ) ]
```

Запускает программу на [MSX BASIC](#) у ученика.

*Пример 1*

```
CALL RUN (1,100)
```

Эта команда запускает программу указанного ученика с заданного номера строки. В этом примере программа запускается с 100 строки у 1 ученика. Если в программе ученика не окажется 100 строки, эта команда вызовет ошибку, сообщение о которой появится на экране ученика. Если программа ученика была уже запущена, то она останавливается и начинается с указанной строки.

*Пример 2*

```
CALL RUN
```

Когда номер строки опускается, программа запускается со своей первой строки. Когда опускается номер ученика, эта команда вызывает запуск программы на [MSX BASIC](#) у всех учеников.

# CALL STOP

Команда преподавателя

Формат:

```
CALL STOP [( <номер ученика>)]
```

Останавливает программу на [MSX BASIC](#).

*Пример*

```
CALL STOP (5)
```

Когда у указанного ученика запущена программа на [MSX BASIC](#), эта команда останавливает её. Когда номер ученика не указан, эта команда вызывает остановку программ на [MSX BASIC](#) у всех учеников. В этом примере останавливается программа у ученика номер 5. Действия этой команды аналогичны действию клавиш `CTRL+STOP`.

Например если ученик находится в режиме ожидания прямого ввода, то при выполнении этой команды будет переведена строка, если в момент передачи команды у ученика выполнялась команда `LIST`, то вывод текста программы будет прерван, и т.д.

# CALL PON

Команда преподавателя

Формат:

```
CALL PON
```

Начинает упорядоченный опрос.

*Пример*

```
CALL PON
```

Преподаватель использует эту команду для начала упорядоченного опроса учеников. Помните, что если этот опрос не активирован, не возможно определить кто из учеников подключён к сети, а также не возможна какая-либо связь между учениками ([CALL TALK](#) и т.д.). При инициализации сети, в том числе при включении компьютера преподавателя, сеть устанавливается в режим опроса.

# CALL POFF

Команда преподавателя

Формат:

```
CALL POFF  
<code>
```

Прекращает упорядоченный опрос.

```
__//Пример//__  
<code>  
CALL POFF
```

Эта команда прекращает упорядоченный опрос учеников. В этом режиме ученики не могут посылать сообщения при помощи команды [CALL TALK](#), а также невозможна всякая связь между учениками.

# CALL ONLINE

Команда ученика

Формат:

```
CALL ONLINE
```

Включает ученика в сеть.

*Пример*

```
CALL ONLINE
```

Эта команда включает ученика в сеть после того, как он был отключён от неё при помощи команды [CALL OFFLINE](#). Если ученик не включён в сеть, какая-либо связь с ним невозможна. При включении компьютер подключается к сети.

# CALL OFFLINE

Команда ученика

Формат:

```
CALL OFFLINE
```

Отключает ученика от сети.

*Пример*

```
CALL OFFLINE
```

Эта команда отключает ученика от сети. Когда компьютер отключён от сети, он не может принять команду от преподавателя. Таким образом, эта команда используется, когда ученик хочет работать сам, без каких-либо прерываний от преподавателя. Чтобы снова включиться в сеть, ученик должен использовать команду [CALL ONLINE](#).

# Ссылки

<http://www.sensi.org/~tnt23/msx/net-1.txt>

<http://www.sensi.org/~tnt23/msx/net-2.txt>

<http://cax.narod.ru/msx/msx-link/doc/MSX-NetBasic-eng.txt>

Category:Network BASIC - MSX Wiki

[https://sysadminmosaic.ru/msx/network\\_basic](https://sysadminmosaic.ru/msx/network_basic)

2023-06-05 19:36

