MCS-52 BASIC Language

Quick Reference
80C32 BASIC Control Computer
Instruction Set 1

ELEKTOR ELECTRONICS

DATASHEET 2/98

COMMANDS		
Command	**Function**	**Example(s)**
RUN	Execute a program	RUN
CONT	CONTinue after a STOP or Control-C	CONT
LIST	LIST program to the console device	LIST LIST 10-50
LIST#	LIST program to serial printer	LIST# LIST# 50
NEW	erase the program stored in RAM	NEW
NULL	set NULL count after carriage return-line feed	NULL
RAM	evoke RAM mode, current program in READ/WRITE memory	RAM
ROM	evoke ROM mode, current program in ROM/EPROM memory	ROM ROM 3
XFER	transfer a program from ROM/EPROM to RAM	XFER
PROG	save the current program in EPROM	PROG
PROG1	save baud rate information in EPROM	PROG1
PROG2	save baud rate information in EPROM and execute program after RESET	PROG2
FPROG	save the current program in EPROM using the INTELligent algorithm	FPROG
FPROG1	save baud rate information in EPROM using the INTELligent algorithm	FPROG1
FPROG2	save baud rate information in EPROM and execute program after RESET, use INTELligent algorithm	FPROG2
STATEMENTS		
Statement	**Function**	**Example(s)**
BAUD	set baud rate for line printer port	BAUD 1200
CALL	CALL assembly language program	CALL 900H
CLEAR	CLEAR variables, interrupts and Strings	CLEAR
CLEARS	CLEAR Stacks	CLEARS
CLEARI	CLEAR interrupts	CLEARI
CLOCK1	enable REAL TIME CLOCK	CLOCK1
CLOCK0	disable REAL TIME CLOCK	CLOCK0
DATA	data to be read by READ statement	DATA 100

MCS-52 BASIC Language

Quick Reference
80C32 BASIC Control Computer
Instruction Set 3

ELEKTOR ELECTRONICS

DATASHEET 2/98

POP	POP argument stack to variables	POP A,B,C
PWM	PULSE WIDTH MODULATION	PWM 50,50,100
REM	remark	REM DONE
RETI	return from Interrupt	RETI
STOP	break program execution	STOP
STRING	allocate memory for strings	STRING 50,10
UI1	evoke User console Input routine	UI1
UI0	evoke BASIC console Input routine	UI0
UO1	evoke User console Output routine	UO1
UO0	evoke BASIC console Output routine	UO0
OPERATORS – DUAL OPERAND		
Operator	**Function**	**Example(s)**
+	addition	1 + 1
/	division	10 / 2
**	exponentiation	2**4
*	multiplication	4*4
–	subtraction	8–4
.AND.	logical AND	10.AND.5
.OR.	logical OR	2.OR.1
.XOR.	logical exclusive OR	3.XOR.2
OPERATORS – SINGLE OPERAND		
Operator	**Function**	**Example(s)**
ABS()	absolute value	ABS(–3)
NOT()	ones complement	NOT(0)
INT()	integer	INT(3.2)
SGN()	sign	SGN(–5)
SQR()	square root	SQR(100)
RND	random number	RND

MCS-52 BASIC Language

Quick Reference
80C32 BASIC Control Computer
Instruction Set 2

READ	read data in DATA statement	READ A
RESTORE	restore READ pointer	RESTORE
DIM	allocate memory for arrayed variables	DIM A(20)
DO	set up loop for WHILE or UNTIL	DO
UNTIL	test DO loop condition(loop if false)	UNTIL A=10
WHILE	test DO loop condition	WHILE A=B
END	terminate program execution	END
FOR-TO-{STEP}	set up FOR-NEXT loop	FOR A=1 TO 5
NEXT	test FOR-NEXT loop condition	NEXT A
GOSUB	execute subroutine	GOSUB 1000
RETURN	return from subroutine	RETURN
GOTO	go to program line number	GOTO 500
ON GOTO	conditional GOTO	ON A GOTO 5,20
ON GOSUB	conditional GOSUB	ON A GOSUB 2,6
IF-THEN-{ELSE}	conditional test	IF A=B THEN A=0
INPUT	INPUT a string or variable	INPUT A
LET	assign a variable or string a value (LET is optional)	LET A=10
ONERR	ONERR or GOTO line number	ONERR 1000
ONTIME	generate an interrupt when TIME is equal to or greater than ONTIME, argument-line number is after comma	ONTIME 10, 1000
ONEX1	GOSUB to line number following ONEX1 when INT1 pin is pulled low	ONEX1 1000
PRINT	PRINT variables, strings or literals, P. is shorthand for PRINT	PRINT A
PRINT#	PRINT to software serial port	PRINT# A
PHO.	PRINT HEX mode with zero suppression	PHO.A
PH1.	PRINT HEX mode with no zero	PH1.A
PHO.#	PHO.# to line printer	PHO.#A
PH1.#	PH1.# to line printer	PH1.#A
PUSH	PUSH expression on argument stack	PUSH 10,A

MCS-52 BASIC Language

Quick Reference
80C32 BASIC Control Computer
Instruction Set 4

LOG()	natural log	LOG(10)
EXP()	e (2.7182818) to the x	EXP(10)
SIN()	returns the sine of argument	SIN(3.14)
COS()	returns the cosine of argument	COS(0)
TAN()	returns the tangent of argument	TAN(.707)
ATN()	returns the arctangent of argument	ATN(1)
OPERATORS – SPECIAL FUNCTION		
Operator	**Function**	**Example(s)**
CBY()	read program memory	P.CBY(4000)
DBY()	read/assign internal data memory	DBY(99)=10
XBY()	read/assign external data memory	P.XBY(10)
GET	read console	P.GET
IE	read/assign IE register	IE=82H
IP	read/assign IP register	IP=0
PORT1	read/assign I/O port 1 (P1)	PORT1=0FFH
PCON	read/assign PCON register	PCON=0
PCAP2	read/assign RCAP2 (RCAP2H:RCAP2L)	RCAP2=100
T2CON	read/assign T2CON register	P.T2CON
TCON	read/assign TCON register	TCON=10H
TMOD	read/assign TMOD register	P.TMOD
TIME	read/assign the real time clock	P.TIME
TIMER0	read/assign TIMER0 (TH0:TL0)	TIMER0=0
TIMER1	read/assign TIMER0 (TH1:TL1)	P.TIMER1
TIMER2	read/assign TIMER0 (TH2:TL2)	TIMER2=0FFH
STORED CONSTANT		
PI	pi = 3.1415926	PI