Instruction set of the AttoBASIC interpreter

Version: 2.1

Control

CONTROL-C.....The control-c keyboard combination halts program operation. Press twice.

CONTROL-S.....The control-s keyboard combination stops the program until another key is pressed

NEW.....New Program EX: NEW

LIST.....List program Ex: LIST

PRINT.....Print value to screen Ex: PRINT A

PRX.....Print hex Ex: PRX 100 results in the output: 64

PRB.....Print binary Ex: PIB INB prints PINB in binary

$ (Dollar Sign).Convert two following characters from ASCII EX: A:=$31

KEY.....Get key from terminal Ex: A := KEY ; or KEY (return) to pause.

EMIT.....Emit value as ASCII character to terminal Ex: EMIT $20 (sends a space).

RND.....Creates an 8-bit random number. Ex: PRI RND [or] A:= RND.

RUN.....Run Program Ex: RUN

IF-THEN.....Control structure Ex: IF A=31 THEN GOTO 100

FOR-TO-NEXT.....Looping structure Ex: see below

GOSUB-RETURN....Program flow control Ex: see below

GOTO LINENUM....Flow Ex: GOTO 100

DELAY [x].....Delay "X" & 10mS. Ex: DELAY 20 delays 200mS

SIZE.....Print the Remaining Bytes Of Program Space To Screen Ex: SIZE

; (Semicolon)...Separate commands on a program line. Ex: TWI ; TWS ; TWA $5C ; TWW $55 ; TWP initializes the TWI interface, asserts a START condition, addresses the slave at address $5C, writes "$55" to it and asserts a STOP condition. [*Note: the semicolon is only valid when embedded in a program line*]

DUMP.....Dump program memory in hex format EX: DUMP

VDUMP.....Dump the contents of the variables [A..Z] Ex: DUMP

EDUMP.....Dump EEPROM memory in hex format EX: EDUMP

END.....Stop execution of program EX: END (this command is not required at end of program)

<backspace>.....Destructive backspace during line editing

SAVE.....Save program to EEPROM Ex: SAVE [*Note: the SAVE command will complain if the program is too big for storage in EEPROM*]

LOAD.....Load program from EEPROM Ex: LOAD

BLDR.....Invoke the boot-loader [*Note: this command uses the AVR's BOOTSZ1:0 fuse bits to determine the location and existence of a boot-loader before jumping to it. An error message is displayed if no boot-loader exists.*]

Operator/Relational

:=.....Set equal to (LET instruction not needed)

=.....Used for evaluation as in IF a = b THEN...)

<>.....Not equal to

>.....Is greater than

<.....Is less than

-.....Subtraction, 8 bit unsigned

+.....Addition, 8 bit

*.....Multiplication, 8-bit

Instruction set of the AttoBASIC interpreter

Version: 2.1

/.....Division, 8-bit
AND.....Logical AND between two 8 bit values
OR.....Logical OR between two 8 bit values
XOR.....Logical Exclusive OR between two 8 bit values
LSL.....Logical shift left
LSR.....Logical shift right
COM.....Compliment (1's compliment or bitwise inversion)
NEG.....Negate (2's compliment)
AOV [x].....Enable arithmetic overflow and underflow detection where x = 1 enables error detection and x = 0 disables error detection. Without [x] is same as x = 0. Defaults to x = 1. Note that when detection is disabled, the result from an arithmetic operation will return the 8-bit result. Expect errors if not careful!

Input Capture

ICG [x].....Initializes ICP mode and sets Input Capture gate time to x[0..7] where x is optional (default 0). Ex: ICG 7 enables ICP registers and sets gate capture time to 1 second.

0 = disables the ICP functi on.	3 = 50mS gate time
	4 = 100mS gate time
	5 = 250mS gate time
1 = 10mS gate time	6 = 500mS gate time
2 = 25mS gate time	7 = 1000mS gate time

ICE [x].....Optionally sets the capture edge. Where x = 0 for falling and 1 for rising (default is 1). Ex: ICE 1 set capture on rising edge.
ICP.....Returns the low byte value and stores the high byte in variable 'Z'. 'Z' is clobbered when executing this command so be aware. Returns an error if there is a 16-bit overflow (and clears 'Z'). Ex: PRX ICP then PRX Z [*Note: the maximum capture frequency depends on the AVR's system clock. Consult the datasheet for specifics*]

Bit I/O

PEEK [Pg],[Of] Read value from data space. Where [Pg] is page number and [Of] is offset into the page. Ex: PRX PEEK $04,$FF reads the byte at $04FF.
POKE [X],[Pg],[Of] Write value [x] to data space. Where [Pg] is page number and [Of] is offset into the page. Ex: POKE A,$01,$00 (POKE VALUE,destination).
[*Note: 1) Variables A-Z may be used, 2) if only [Pg] is specified, it is used as [Of], the offset into page zero*]

Note: For the following Port I/O commands, substitute [p] for the port value (A..D) if relevant for the MCU AttoBASIC has been compiled for. Examples are show for each command.
OD[p].....Output data direction register DDR[p] EX: ODB $FF
ID[p].....Input from data direction register DDR[p] EX: J:= IDC
SD[p].....Set bit in data direction register DDR[p] EX: SDD 3
CD[p].....Clear bit in data direction register DDR[p] EX: CDA 3

OP[p].....Output PORT[p] EX: OPA $1A
SB[p].....Set bit on PORT[p] EX: SBB 3

CB[p].....Clear bit on PORT[p] EX: CBC 3

IN[p].....Input from PIN[p] EX: J:= INAC
IB[p].....Input bit from PORT[p] EX: IF IBD 2 THEN GOTO 100

Pulse Width Modulator

PWM8[x].....Pulse width modulation 8 bit on OC0A [or] OC1A PIN EX: PWM 17 *[Note: the default is OC1A]*

PWE [x].....PWM extended 10 bit pwm on OC1A pin EX: PWE 2,00 *[Note: the PWE command may not be enabled if the compiled MCU timer does not support it, i.e. if OC0A is selected as the PWM channel at compile time, this command is disabled]*

PWO.....PWM on OC1A [or OC0A] PIN OFF (does not affect any data direction register).

Analog Comparator

ACO.....Analog comparator output EX: IF ACO THEN PRINT A. Prints a if analog comparator output is high

Analog to Digital Converter

ADR [x].....ADR [x] initializes the ADC and sets the ADC reference to Internal or External. x = 0 for INT and x = 1 for EXT. Without [x] is same as x = 0, int. ref. Ex: ADR 1 selects external Vref for ADC. *[Note: this command must be executed before obtaining readings from the ADC]*

ADC [x].....8-bit ADC conversion EX: PRX ADC [or] PRX ADC 9 [or] PRX ADC 15. Without [x] is same as x = 0 *[Refer to the appropriate AVR data sheet for valid ADC channel numbers as some AVR's support reading the on-chip temperature and Vref. This command does not error check 'x'.]*

DS Interface

DSDATA.....Send a byte over the DS Interface as data EX: DSDATA A

DSCOMMAND.....Send a byte over the DS Interface as a command EX: DSCOMMAND C

DSREAD.....Read a byte from the DS Interface EX: PRX DSREAD

DDS (Direct Digital Synthesis) Interface

DDS [x].....Outputs a frequency on the defined port pin at the 6-BCD-digit frequency held in the X/Y/Z variables. X = 0 to disable DDS and x = 1 to enable [X/Y/Z set first]. Without X same as 0 [disable]. The DDS sample frequency is set to twice the Interrupt service routine's duration, which is 5uS. Therefore, the output frequency range will be 0 to 25KHz in 1Hz steps. Ex (as separate commands): X:= 01, Y:= 23, Z:= 45, DDS 1 will emit a 12.345KHz frequency on the DDSOut pin.

Instruction set of the AttoBASIC interpreter
Version: 2.1

SPI Interface

SPM [x].....MUST be called first to initializes the SPI hardware to operate in Mode [0..3]. Without [x] is same as x = 2; Master, Mode 2, F_CLK/16, MSB first. (Refer to the AVR data sheet for explanation of mode #'s)

SPO [x].....Optionally set MSB/LSB data order where x = 0 for MSB and x = 1 for LSB.

SPC [x].....Optionally set SPI clock to [0.15]. (Refer to the AVR data sheet for explanation of mode #'s)

SPW [x].....Write a byte to SPI. Note that SPI_SS pin is set low when this command is executed and not restored so user must toggle the pin high with the SPS command.

SPR.....Read a byte from SPI. Note that SPI_SS pin is set low when this command is executed and not restored so user must toggle the pin high with the SPS command.

SPS [x].....Set the SPI_SS pin to logic level of [x]. Defaults to '1'

TWI Interface

TWI [x].....TWI must be called first to initialize the TWI interface. X = 0 for 400Kbps and x = 1 for 100Kbps clock. Without [x] is same as x = 0. Defaults to Master @ 400Kbps with PORT pull-ups enabled. [*Note: A 6.4MHz clock is required to operate the TWI at 400K. Therefore using a clock below 6.4MHz will always initialize the TWI interface at 100K regardless of the value given for "x". If it is desired to use alternate pull-ups, disable the PORT pull-ups by clearing the SCL/SDA pins in the PORT register. Ex (as separate commands): CBC 4, CBC 5].*

TWS.....Assert a START condition on the bus. When the TWI interface is initialized, a START condition is asserted. Returns with the bus status on the stack. However, the user must re-assert a START condition after a STOP condition to ready the bus for the next message sequence.

TWP.....Assert a STOP condition on the bus. The user must assert a STOP condition after the last message byte has been sent to or received from the slave or to abort a transfer in progress.

TWA [x].....TWA sends the slave address to the bus. Returns with the bus status on the stack. Ex 1: TWA $A0 selects slave address $A0 for writing. Ex 2: A:= TWA $A0 selects slave address $A0 for writing and returns the bus status in variable A. [*Note: This command should be used after issuing a START condition to send the desired slave address. The user must insure bit 0 of the slave address contains the R(ead) or W(rite) indicator bit AND'ed or OR'ed with the 7-bit slave address before sending. The address may need to be left-shifted one bit position*]

TWW [x].....TWW sends a byte to the bus. Returns with the bus status on the stack. Ex 1: TWW B eend the data held in variable B to the previously selected slave for a write operation. Ex 2: A:= TWW $A0 sends $A0 to the slave for writing and returns the bus status in

variable A. . [Note issue this command after a "TWA [x]" (SLA+W) has been issued and acknowledged by the slave].

TWR [x]..... Receives a byte from the TWI bus and places it onto the stack.. x = 0 to signal to the slave that this is the last byte to receive, x = 1 to signal to the slave there is more data to receive. Without [x] is same as x = 1. Ex: A:= TWR 0 receives a byte, signal to the slave that no further data is requested and returns the data in variable A. [Note issue this command after a "TWA [x]" (SLA+R) has been issued and acknowledged by the slave].

TWB..... Queries the TWI status register for the last detected condition of the bus. [*Note: the byte returned is right-shifted 3 bit positions. If a STOP condition has been detected, $80 is returned to indicate so*]. Ex: A:= TWB (if A = 3 then SLA+W has been transmitted and an ACK received).