Instruction set of the AttoBASIC interpreter

Version: 2.22

Control

CONTROL-C	The control-c keyboard combination halts program operation. Press twice. NOTE: on some keyboards, use the "BREAK" key.
CONTROL-S	The control-s keyboard combination stops the program until another key is pressed (not another ^S).
HELP	Display a list of commands available on the current build version.
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: PRB INB prints PINB in binary
"str"	Print string enclosed in quotes Ex: PRI "123~456~" prints "123" followed by a CR/LF then "456" followed by a CR/LF. *[Note: **1) strings eat program space!** 2) the "~" character starts a new line 3) in immediate mode, max string size is 57 characters]*
$ (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
FREe	Print the remaining bytes of program space to screen Ex: FREE *[Note: This command was "SIZE" in previous versions.]*
; (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 and not in immediate mode*]
# (pound sign)	The line interpreter ignores all characters after "#" until a CR or LF is found. This allows one to add comments to programs edited externally and uploaded via the terminal interface. [*Note: this is equivalent to REM but the characters are not stored in program memory.]*
<backspace>	Destructive backspace during line editing.
OSC [x]	Set or read the AVR's Oscillator Calibration Register. EX: "PRI OSC" prints the OSCCAL value while "OSC $A8" sets the OSCCAL register to 0xA8.
DUMp	Dump RAM 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)
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.*]
LDD	Load Default program into program memory. Note: this command is only available for use with the AVR Data Recorder.
RST	Causes a system reset using the watchdog "System Reset" mode.

Instruction set of the AttoBASIC interpreter

Version: 2.22

Relational Operators

=	Used for evaluation as in IF a = b THEN...
!=	Not equal to (was "<>" in previous versions)
>	Is greater than
<	Is less than

Note: The relational operators return "1" for a true comparison and "0" for false comparison. This behavior is different from previous versions.

Arithmetic Operators

:=	Set equal to (LET instruction not needed)
-	Subtraction, 8-bit unsigned
+	Addition, 8-bit unsigned
*	Multiplication, 8-bit unsigned
/	Division, 8-bit unsigned
%	Modulus, 8-bit unsigned
AND (or '&')	Bitwise AND between two 8-bit values
OR (or '|')	Bitwise OR between two 8-bit values
XOR (or '^')	Bitwise XOR between two 8-bit values
LSL	Logical shift left
LSR	Logical shift right
COM	Compliment (1's compliment or bitwise inversion)
NEG (or '!')	Negate (2's compliment, was "NEG" in previous versions)
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 = 0. Note that when detection is disabled, the result from an arithmetic operation will return the 8-bit result. **Also be aware that if AOV is enabled, the IF statement may cause unexpected errors.** Expect errors if not careful!

Input Capture

ICG [x]	Initializes ICP mode on the ICP1 pin 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 function.	4 = 100mS gate time
1 = 10mS gate time	5 = 250mS gate time
2 = 25mS gate time	6 = 500mS gate time
3 = 50mS 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 [P] [O]	Read value from data space. Where [P] is page number and [O] is offset into the page. Ex: PRX PEEK $04,$FF reads the byte at $04FF.
POKe [X] [P] [O]	Write value [x] to data space. Where [P] is page number and [O] is offset into the page. Ex: POKE A,$01,$00 (POKE VALUE,destination). [*Note: 1) Variables A-Z may be used, 2) if only [P] is specified, it is used as [O], the offset into page zero*]

Instruction set of the AttoBASIC interpreter
Version: 2.22

Note: For the following Port I/O commands, substitute [p] for the MCU specific port value (A..D, etc.) and [x] for the value to be read or written. Examples are show for each command.

OD[p] [x]	Output data direction register DDR[p] EX: ODB $FF
ID[p] [x]	Input from data direction register DDR[p] EX: J:= IDC
SD[p] [x]	Set bit in data direction register DDR[p] EX: SDD 3
CD[p] [x]	Clear bit in data direction register DDR[p] EX: CDA 3
OP[p] [x]	Output PORT[p] EX: OPA $1A
SB[p] [x]	Set bit on PORT[p] EX: SBB 3
CB[p] [x]	Clear bit on PORT[p] EX: CBB 3
XB[p] [x]	XOR (toggle) bit on PORT[p]EX: XBC 3
IN[p] [x]	Input from PIN[p] EX: J:= INA 2
IB[p] [x]	Input bit from PORT[p] EX: IF IBD 2 THEN GOTO 100

*The following command is for use with the **AVR Data Recorder***

DIG [x]	Switches the N-MOSFET on or off; X = 0 to turn off, X = 1 to turn on.

Pulse Width Modulator

PWM8[x]	Pulse width modulation 8-bit on OC0A [or] OC1A PIN. The output voltage is (X * (VCC / 255)) volts. EX: PWM 17 *[Note: the default is OC1A].*
PWE [x] [y]	PWM extended 10 bit PWM on OC1A pin where x = bit 9:8 and y = bit 7:0. EX: PWE 2,00 sets the PWM duty cycle to 50%. *[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 the port's data direction register).

Analog Comparator

ACO	Analog comparator output EX: IF ACO THEN PRINT A. Prints the value of "A" if analog comparator output is high
ACR [x]	Sets Analog Comparator Reference voltage to "X". The output voltage is (X * (VCC / 255)) volts. This is the same command as "PWM x" and is implemented for use with the *AVR Data Recorder but can be used if RC filtered and fed to the Analog Comparator's "+" input*.
ACS [x]	Select the source for the Analog Comparator's "-" input. When X = 0, ADC0 is selected. When X = 1, ADC1 is selected. When X = 2, AIN- is selected. Default is X = 2.
ACI [x]	When X = 1, the Analog Comparator's interrupt is enabled and triggers on a change of state. When X = 0, it is disabled. This command is intended to support the "*SLP 0*" command. Note that the analog comparator interrupt can be used with the AIN-, ADC0 or ADC1 sources selected with the *ACS* command.

Analog to Digital Converter

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 0 [or] A:= ADC 9 [or] PRX ADC 15. *[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'.]

*The following commands are for use with the **AVR Data Recorder***

ADG [n] [g]	*Set the analog gain for channel "N" to "G". "N" is "0" or "1" for channel 0 or 1. G = 0 for gain of 0.01x, X = 1 for gain of 0.1x, X = 2 for gain of 1x and X = 3 for gain of 10x. Ex: ADG 0,3 sets analog channel 0 to a gain of 10x.*
ADS [x]	*Select channel "0" or channel "1" as source for TRMS to DC conversion. Ex: ADS 1 selects channel 1 as the source for TRMS conversion. [Note: the AD536 can convert AC or DC signals.]*
ADU [x]	*Select the True RMS or dBV output of the AD536. X = 0 for TRMS, x = 1 for dBV. Ex: ADU 1 selects dBV readings.*

DS Interface

DSD	Send a byte over the DS Interface as data EX: "DSD $AA" sends the value of $AA.
DSC	Send a byte over the DS Interface as a command EX: "DSC C" sends the value contained in the variable "C".
DSR	Read a byte from the DS Interface EX: "PRX DSR"

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.

***Notes: 1)** the accuracy of the output frequency is directly related to the accuracy of the CPU's timebase. 2) Due to interrupt servicing latency, some jitter will be seen on the output signal.*

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...7]. (Refer to the AVR data sheet for explanation of SPI clock dividers)
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 sends 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).

Real-time Counter

The Real-time Counter commands are used to access an implementation of a 32-bit internal counter that increments sequentially after a preset period of time has occurred.

RTI [n] — Sets the Real-time counter increment interval, the resolution, to "N", where: N = 0 for 1mS (default), N = 1 for 10mS, N = 2 for 100mS. Ex: "RTI A" will set the interval to the value stored in variable "A".

RTR — Resets the Real-time counter to zero.

RTP — Assign or print the 32-bit value of the Real-time timer. When assigned to a variable, nothing is printed to the console (see notes). When printed to the console, the full 32-bit value (10 digits) is printed (this option is intended to be used when capturing time-referenced data via the console and importing into a spreadsheet or the like). Ex: A: = RTP (assign low 8-bits to variable "A"). Ex: RTP (prints the 10-digit value or the RTC to the console).

NOTES: 1) *The RTC has an internal resolution of 1mS.* ***2)*** *There may be an intrinsic error*

in the Real-time counter, which is dependent on the master CPU clock frequency used. At CPU clock frequencies of 4, 8, 16 and 20MHz, a typical error of +0.00% will be seen. Other frequencies may yield different errors. ***3)*** *At 1mS ("RTI 0"), the maximum count will yield ~49.71 days before "rolling over" to "0". At 100mS ("RTI 2"), the maximum count will yield ~4971 days before "rolling over" to "0".* ***4)*** *The "**RTP**" command can be used to assign the value held in the lower 8-bits of the Real-time Counter to a variable, thus time differences of up to 25.50 seconds can be directly measured.* ***5)*** *Using the "**PEEk**" command gains access to all 32-bits of the counter (stored in RAM) if needed (locate "DSEG RTCReg" in the "map" file; stored as RTCReg[3:0]). 6) One must be aware that when assigning and comparing two values held in variables sampled from the RTC at different times, the second reading may have rolled over from 255 to 0 and be numerically less than the 1st sample. The "AOV" setting will not flag this.*

Data File

The Data File routines use a serial EEPROM attached to the SPI interface. The SS pin is used as the chip-select. Regardless of the device's internal page size, AttoBASIC pages are 256 bytes long with a maximum of 256 pages (16-bit addresses = 65,536 bytes). One obvious use for this command set is as a chart recorder.

DFR [a] [p]	Read data from Data File at address [a] of page [p]. EX: "DFR $80 4" reads the value from address $80 of page 4 (or 0x0480).
DFW [d] [a] [p]	Write data [d] to Data File at address [a] of page [p]. EX: "DFW A 0 8" writes the value contained in variable "A" to address 0 of page 8 (or 0x0800).
DFL [d]	Log (write) data [d] to Data File and increment the internal address counter. EX: "DFL A" writes the value contained in variable "A" to the data file. Without [d] is the same as d = 0.
DFX	Reset the internal data file address counter to "0". The address counter is also reset to "0" upon a hardware RESET or invoking the "RST" command.
DFF [n]	Set Data file dump stream format to [n] values per record. EX: "DFF 4" sets the dump format to four (4) values per record thus allowing data recording for multiple values per record. Without [n] is the same as n = 1. Note: there is no restriction on [n] except N != 0.
DFD [p] [n]	Dump sequential data stream from Data File starting at address 0 of page [p] for [n] pages. EX: "DFS $10 1" prints 1 page of sequential data starting at address 0 of page $10 (or 0x1000). If the DFF command has specified N > 1 then each data value is separated by comma and each record by a CR/LF combination (ex: "0,1,2,3[cr/lf]"). This makes the data output CVS compliant for easy capture.
DFI [p] [n]	Data File Initialize [n] pages ("0" inclusive) starting from page [p]. EX: "DFI 2 3" initializes 3 pages of the Data File staring at page 2. Without [n] erases one page. Without [p] and [n] starts erasure at page zero for one page. *Note: unless set with the DFB command, "0" is the fill value used.*
DFV [d]	Set the value to be used when initializing the Data File with the DFI command. EX: "DFV $55" sets the initialization value to "$55". Without [x] is the same as "0".

NOTES: ***1)*** *There is no test for the existence of an attached device.* ***2)*** *This version of software does not support additional SPI peripherals when the Serial EEPROM routines are enabled 3**)** It is the user's responsibility to insure the page number specified is within the attached device's range.*

DHTxx Temperature and Humidity Sensor Interface

Instruction set of the AttoBASIC interpreter

Version: 2.22

The DHTxx sensor interface commands are used to read the temperature and humidity from DHT11, DHT22 and compatible devices.

DHT	Acquire the temperature reading from the DHTxx sensor. The result is selectable between Fahrenheit and Celsius using the DHU command. For Fahrenheit measurements, the DHT command outputs 0 to 176 degrees (the sign is always ignored). For Celsius measurements, the DHT command outputs 0 to 80 degrees when the sign is ignored or –40 to 80 degrees when the sign is recognized (bit 7 is the sign bit).
DHH	Acquire the humidity reading from the DHTxx sensor, which is in % Relative Humidity between 0 and 100.
DHR	Return the availability status of the DHTxx sensor. "1" means "busy" (less than 2 seconds since last read), while "0" means **R**eady to read.
DHU [x]	Set temperature unit. X = 0 for Fahrenheit (default) and X = 1 for Celsius. Without [x] returns the state of the setting.
DHS [x]	Recognize or ignore signed temperature readings. X = 0 to ignore (default) and X = 1 to recognize. Without [x] returns the state of the setting.

NOTES: ***1)*** *The DHT sensor routines use the pin-change interrupts, therefore, the DHT data pin must be attached to a pin supporting pin-change interrupt. Consult the AVR's data sheet for further information. For the ATMega32U4, this is restricted to PORTB.* ***2)*** *An internal timer is monitored by the DHT routines to insure that at least 2 seconds has elapsed between reads. If not then the previous reading will be returned.* ***3)*** *The integer and decimal values for temperature and humidity are stored in RAM at locations "RT_I" (integer temperature), "RT_D" (decimal temperature), "RH_I" (integer humidity), "RH_D" (decimal humidity). Consult the appropriate map file for details (reference DSEG).* ***4)*** *An error will issue for a non-responsive DHT sensor.* ***5)*** *An error will issue for a data and checksum mismatch.*

Low-power using Sleep

SLP [x]	If "x" is "0" then the AVR executes the low-power sleep instruction and waits for a user-enabled event to occur. If "x" is greater than zero then the AVR enters sleep and uses the watchdog timer as a "one-shot" time delay event source. Without [x] is the same as x = 0. *Refer to the* ***Watchdog Timer*** *section of the specific AVR datasheet for valid timeouts for "x" values greater than zero.*

NOTE: ***1)*** *This instruction can be used instead of DELay if low-power is needed but precision timing is not.* ***2)*** *The AVR's "idle mode" is always used.* ***3)*** *Any supported interrupt source may be used as the event trigger.* ***4)*** *USART interrupts are not masked and will trigger an event.* ***5)*** *Proper use of the sleep mode requires that the user enable the desired interrupt source(s) before executing this instruction.* ***6)*** *Using the* ***POKE*** *statement allows programming the desired Interrupt Control Register and Interrupt Mask Register.* ***7)*** *When invoked, the watchdog timer is implemented in* ***"Interrupt Mode"****, however, the prescaler value of "0" is not available.* ***8)*** *If the RTC and/or DDS routines are enabled then the OCRxA interrupt source for each TIMER used (see source code for specific timer/register) is unavailable. However, the other OCRxB/C/D/E/F interrupts sources are available.*