Picaxe Basic

Digital Input/Output	
high	Switch an output pin on
low	Switch an output pin off
toggle	Change an output pin between on and off
sound	Generate simple sounds
play	Play a pre-defined musical tune
tune	Play a sequence of musical notes
servo	Control a servo
servopos	Set the position of a servo
irin	Receive an infra-red command
irout	Generate an infra-red command
rfin	Receive data from an RF receiver module
rfout	Send data to an RF transmitter module
pwmout	Generate a continuous PWM signal
pwmduty	Set the duty ratio of a PWM signal
if pin	Respond to the state of an input signal
count	Count pulses detected on an input pin
pulsin	Measure the length of a pulse on an input pin
pulsout	Generate a pulse on an output pin
input	Set a pin for receiving input
output	Set a pin to produce an output signal
reverse	Toggle a pin between input and output use
pullup	Set internal pull-ups for input pins
inputtype	Select the electrical characteristics of an input pin
button	Detect and debounce a button push
Analogue Input/Output	
touch	Read a touch sensor status
touch16	Read a touch sensor signal level
readadc	Read an analogue input
readadc10	Read a high-resolution analogue input
readtemp	Read a DS18B20 temperature sensor
readtemp12	Read a DS18B20 temperature sensor to 0.625C accuracy
daclevel	Set an analogue output level
dacsetup	Configure analogue output
readdac	Read the analogue output
readdac10	Read the analogue output using highest resolution
Serial Interfacing	
debug	Update the debug display screen
sertxd	Send serial data out through the Serial Out pin
serrxd	Receive serial data through the Serial In pin
serin	Receive serial data through an input pin
serout	Send serial data out through an output pin
hserin	Receive serial data through the High-Speed Serial In pin
hserout	Send serial data out through the High-Speed Serial Out pin
hsersetup	Configure the High-Speed Serial interface
Interrupts and Multi-Tasking	
setint	Set the input conditions which cause an interrupt
setintflags	Specify the events which cause an interrupt
restart	Restart a program task
resume	Resume executing a suspended program task
suspend	Suspend a program task

Advanced I/O Interfacing	
hi2cin	Read data from an I2C device
hi2cout	Write data to an I2C device
hi2csetup	Configure the I2C mode and the I2C bus
kbin	Read key press data from a PS/2 keyboard
kbled	Control the LEDs on a PS/2 keyboard
owin	Read data from a 1-wire device
owout	Write data to a 1-wire device
readowsn	Read the serial number of a 1-wire device
hspiin	Read data using the High-Speed SPI interface
hspiout	Write data using the High-Speed SPI interface
hspisetup	Configure the High-Speed SPI interface
shiftin	Read data from an SPI device
shiftout	Write data to an SPI device
srlatch	Configure the hardware SR Latch
srreset	Reset the hardware SR Latch
srset	Set the hardware SR Latch
uniin	Read data from a UNI/O device
uniout	Write data to a UNI/O device
hpwm	Generate PWM output signals
hpwmduty	Set the duty ratio of HPWM generated signals
Program Flow Control	
goto	Continue program execution from a named label
branch	Branch to one of a number of named labels
for	Repeat a loop a certain number of times
next	Indicate the end of a 'for' command loop
do	Repeat a loop until a certain condition is met
loop	Indicate the end of a 'do' command loop
if	Conditionally execute program code
else	Alternative code execution for an 'if' command
endif	Indicate the end of an 'if' command
select	Select which section of program code to execute
case	Define a section of program code for a 'select' command
endselect	Indicate the end of a 'select' command
gosub	Call a subroutine
return	Return from a subroutine
end	Terminate program execution
reset	Reset the PICAXE and restart program execution
stop	Stop the program code from continuing
on goto	Continue at one of a number of named labels
on gosub	Select one of a number of subroutines to call
if bit	Conditionally execute code depending on a variable bit setting
exit	Exit from a 'do' or 'for' command loop
Time Delays	
pause	Delay for a number of milliseconds
pauseus	Delay for a number of microseconds
nap	Sleep for a short period of time
sleep	Sleep for a period of time
doze	Reduce power consumption for a short period of time
hibernate	Reduce power consumption for a period of time
wait	Delay for a number of seconds
time	Use the elapsed time counter

Variables	
symbol	Give a name to a variable or number value
let	Perform a mathematical operation
inc	Increment a variable's value by one
dec	Decrement a variable's value by one
swap	Swap the values of two variables between each other
bcdtoascii	Convert a BCD value to its ASCII (text) representation
bintoascii	Convert a numeric value to its ASCII (text) representation
lookdown	Find an item in a list of values
lookup	Select an item value from a list
random	Set a variable to a random value
clearbit	Clear a bit within a variable
setbit	Set a bit within a variable
togglebit	Invert a bit within a variable
peek	Get a value from PICAXE memory
poke	Set a value in PICAXE memory
get	Get a value from scratchpad memory
put	Set a value in scratchpad memory
read	Read a value from internal data EEPROM
write	Set a value in internal data EEPROM
eeprom	Specify values to be loaded to internal data EEPROM
readtable	Get a value from the data table
table	Specify values for the data table
tablecopy	Copy data table values into variables
Directives	
#com	Set the serial/USB COM port for downloading.
#define	Define a name to control conditional compilation
#else	Alternative program code include for #ifdef and #ifndef directives
#endif	Terminate #ifdef and #ifndef commands
#endmacro	End a macro
#endregion	End a region
#endrem	Terminate a previous #rem directive
#error	Force a compilation error to be produced
#freq	Specify the current operating speed of the PICAXE being downloaded into
#gosubs	Specify the number of GOSUBs allowed
#if	Only include program code if a #define value is defined
#ifdef	Only include program code if a #define name is defined
#ifndef	Only include program code if a #define name is not defined
#include	Include program code from another source file
#macro	Define a macro
#no_data	Do not download data EEPROM values
#no_debug	Disable debug commands in a program
#no_end	Do not include a terminating 'end' command

Directives	
#no_table	Do not download data table values
#picaxe	Specify the PICAXE the program code is for
#region	Define a code region
#rem	Treat subsequent program lines as comments
#revision	Specify the revision number in a program slot
#sim	Specify simulation model to use
#simspeed	Set speed of simulation
#simtask	Specify program task to simulate
#slot	Specify the program slot the program code will be downloaded into
#terminal	Set Terminal display baud rate
#undefine	Remove a name previously created with #define
preprocessor	[Pre-processor substitution constants]
Advanced PICAXE Configuration	
peeksfr	Read data from an internal control register
pokesfr	Write data to an internal control register
calibfreq	Adjust the operating speed of the PICAXE
setfreq	Set the operating speed of the PICAXE
disablebod	Enable low-voltage operation
enablebod	Disable low-voltage operation
readinternaltemp	Read the internal temperature sensor
readfirmware	Obtain PICAXE firmware information
readrevision	Obtain the #revision details for a program slot
readsilicon	Obtain information on the PICAXE chip type
adcconfig	Configure analogue input operation
adcsetup	Configure analogue input channels
calibadc	Read the internal voltage reference value
calibadc10	Read the internal voltage reference value with highest resolution
compsetup	Configure analogue comparators
fvrsetup	Configure the on-chip voltage reference
disconnect	Prevent the PICAXE accepting program downloads
reconnect	Enable the PICAXE to accept program downloads
hintsetup	Configure hardware interrupt input pins
disabletime	Prevent the elapsed time variable incrementing
enabletime	Enable the elapsed time variable to increment
settimer	Configure the internal timer
tmr3setup	Configure internal Timer 3
booti2c	Copy external program slot to internal program memory
run	Run the program in another program slot
pop	Pop a byte from the stack
popram	Pop variables from the RAM stack
push	Push a byte to the stack
pushram	Push variables to the RAM stack