

The EZ-Builder Script Manual.



This is an online version of the EZ-Script manual for easy reference. It is a copy of what can be found under the "EZ-Script Help" tab inside of a controls script editor inside EZ-Builder. There are two versions of the manual in this document... the full one page manual, and a broken down version arranged in alphabetical order for your convenience. Use this online manual when you want to reference it, but do not have EZ-Builder on your device.



EZ-Script Functions:

Sleep (milliseconds)

Pauses for specified milliseconds

Example sleeps for 1 second: *Sleep(1000)*

SleepRandom (lowMilliSec, highMilliSec)

Pauses for a random millisecond delay between the 2 provided values

Example: *SleepRandom(1000, 5000)*

Servo (servoPort, position)

Move servo to the specified position

Servo position is between 1 and 180

Example: *Servo(D14, 25)*

SetServoMin (servoPort, position)

Set the minimum limit that this servo can ever move to

Servo position is between 1 and 180

Example: *SetServoMin(D14, 40)*

SetServoMax (servoPort, position)

Set the maximum limit that this servo can ever move to

Servo position is between 1 and 180

Example: *SetServoMax(D14, 100)*

PWM (digitalPort, speed)

Set the PWM (Pulse Width Modulation) to the desired duty percentage cycle

This simulates voltage on the specified pin (Between 0 and 5v)

PWM Value is between 0 and 100

Example: *PWM(D14, 90)*

GetPWM (digitalPort)

Gets the PWM (Pulse Width Modulation) of specified port

PWM is between 0 and 100

Example: $\$x = \text{GetPWM}(D14)$

PWMRandom (digitalPort, lowSpeed, highSpeed)

Set the PWM (Pulse Width Modulation) to a random percentage duty cycle

This simulates voltage on the specified pin (Between low and high percentage value, scaled between 0 and 5 volts)

The value is between 0 and 100

Example: *PWMRandom(D14, 10, 90)*

ServoSpeed (servoPort, speed)

**ACos(value)**

Returns the math ACos() function (also called ArcCos)

Example: $x = \text{Acos}(27)$

Abs(value)

Returns the absolute value of a number

Converts a negative into a positive number

Example: $x = \text{Abs}(-22)$

ADC_Wait (adcPort, higher/lower/equals, value, [delay ms])

Wait until ADC port is higher or lower than specified value

The optional parameter Delay MS is the millisecond delay for checking. This value determines the delay between checks.

Example: `ADC_Wait(ADC0, HIGHER, 50)`

Example: `ADC_Wait(ADC0, HIGHER, 50, 50)`

ADC_Wait_Between (adcPort, low, high, [delay ms])

Wait (pauses script) until ADC port is between the specified values. Soon as the ADC port is between the low and high values, it will stop waiting.

The optional parameter Delay MS is the millisecond delay for checking. This value determines the delay between checks.

Example: `ADC_Wait_Between(ADC0, 20, 50)`

Example: `ADC_Wait_Between(ADC0, 20, 50, 50)`

AppendArray(variable, value)

Appends the value to an existing array. This grows the size/length of the array to hold the new value.

Example: `AppendArray($myArray, 10)`

Example: `AppendArray($myArray, "banana")`

ASin(value)

Returns the math ASin() function (also called ArcSin)

Example: $x = \text{ASin}(27)$

AX12Led(id, on/off)

Controls the LED status of an Dynamixel AX-12 Servo by its ID on port D5. Check the port modes in the respective section near the end of this document.

Example: `AX12LED(1, on)`

Example: `AX12LED(1, off)`

References**ADC Ports**

The ADC Ports are labeled on the EZ-B as A0 to A7. They are Analog Input ports, which read the voltage of the incoming data between 0 and 5 volts. Consult the Learn Section of our website for more information on Port Types.

B.)



Browser(url)

Launches the default web browser with the specified URL.

Example: *Browser("http://www.google.com")*

**ClearVariables()**

Clear all variables. Variables are not cleared between projects. You may wish to call this function in your initialization script - unless your project is using variables from another previous project.

Example: *ClearVariables()*

CheckForUpdate()

Checks the EZ-Robot server if there is a newer version of EZ-Builder available.

This requires an internet connection.

Example: *\$x = CheckForUpdate()*

CloseControl()

Used for mobile devices and the Interface Builder only, this command will close the current control, the same as pressing the BACK button on your device.

Example: *CloseControl()*

ComAvailable (Port)

Returns the number of bytes/characters in the input buffer of the specified serial communication port on the local PC. The port must be open before this command can be called.

Example: *\$size = ComAvailable("com1")*

ComClearInput(Port)

Clear the input buffer. The port must be open before this command can be called.

Example: *ComClearInput("com1")*

ComClose(Port)

Close the specified serial communication port on the local PC.

Example: *ComClose("com1")*

ComOpen(Port, Baud Rate)

Open the specified serial communication port on the local PC. When opened, this will also begin buffering incoming data which can be read. The input buffer is 128KB.

Example: *ComOpen("com1", 9600)*

ComRead(Port, Bytes to Read)

Read specified number of characters from the input buffer of the specified serial communication port on the local PC. The port must be open before this command can be called.

Example: *\$variable = ComRead("com1", 15)*

ComReadAll(Port)

Read all available string data from the input buffer of the specified serial communication port on the local PC. The port must be open before this command can be called.

Example: *\$variable = ComReadAll("com1")*



DefineArray(variable, size, [default value])

Creates an array of the variable to the specified size.

Optionally, you may also pass a value that will be used for the default values.

Example: *DefineArray(\$myArray, 10)*

Example: *DefineArray(\$myArray, 10, 2)*

Digital_Wait (digitalPort, on/off/true/false, [delay ms])

Wait until the digital port status has changed

The optional parameter Delay MS is the millisecond delay for checking. This value determines the delay between checks.

Example: *Digital_Wait(D12, ON)*

Example: *Digital_Wait(D12, ON, 50)*

Down([milliSeconds])

Using the servo port settings from a Movement Panel Control, this will lower your drone robot up

Optionally, you can specify the number of milliseconds to turn.

Example #1: *Down()*

Example #2: *Down(1000)*

DroneEmergency()

Tell your flying drone to reset from emergency or power down when flying.

This command should be added a button on the joystick so you may stop the drone so it does not get away or in danger.

DumpVariables()

Prints a list of all variables and their respective values separated by CRLF.

This is used for the TCP Server clients which require a collection of variables.

Example: *DumpVariables()*

References

Digital Ports

These ports are used to turn on and off devices with voltage. The digital ports can also be used for detecting if the input voltage is in an On or Off state. For output, the digital ports may control servos, transmit PWM (Pulse Width Modulation) and send serial data. Consult the Learn Section of our website for more information on the different modes of Digital Port Types.

D0

D1

D2

D3

D4

D5

D6

D7

D8

D9

D10

**Else**

Condition tree must be closed with an ENDIF

Example:

```
If (GetDigital(D0) = 1)
```

```
Print("Yes")
```

```
Else
```

```
Print("No")
```

```
EndIf
```

Exec(EXE/Bat File, [parameters])

Executes a windows application or batch file. The second parameter is a list of optional parameters

Example: `Exec("C:\Windows\notepad.exe")`

Example: `Exec("C:\Windows\notepad.exe", "C:\MyFile.txt")`

**FileExists(filename)**

Returns a 1 or 0 if the specified file exists.

Example: *\$fileExists = FileExists("c:\temp\mylog.txt")*

FileDelete(filename)

Deletes a file on your computer

Example: *FileDelete("c:\temp\mylog.txt")*

FileReadAll(filename)

Returns the entire contents of the specified file.

Example: *\$contents = FileReadAll("c:\temp\mylog.txt")*

FileReadChar(filename)

Returns the next character in the specified file

Example: *\$char = FileReadChar("c:\temp\mylog.txt")*

FileReadClose(filename)

Closes the file from reading.

This must call must be performed before writing to the file. Once you begin reading from the file, the file is OPEN. Closing the file will reset to the start once you begin reading again.

Example: *FileReadClose("c:\temp\mylog.txt")*

FileReadEnd(filename)

Returns a 1 or 0 if the file has reached the end.

Example: *\$fileEnd = FileReadEnd("c:\temp\mylog.txt")*

FileReadLine(filename)

Returns the next line of the specified filename.

Example: *\$line = FileReadLine("c:\temp\mylog.txt")*

FileReadLineRandom(filename)

Returns a random line within the specified file

Example: *\$randomLine = FileReadLineRandom("c:\temp\mylog.txt")*

FileReadReset(filename)

Resets the file to the beginning.

If you read to the end of a file, this function must be called to reset reading from the beginning of the file.

Example: *FileReadReset("c:\temp\mylog.txt")*

FileWrite(filename, contents)

Appends text to the specified file. This does not append a new line.

Example: *FileWrite("c:\temp\mylog.txt", "My Variable: " + \$x)*

**GetADC(Port)**

Returns the 8 Bit ADC value of the specified port

Example: $\$x = \text{GetADC}(\text{adc0})$

GetADC12(Port)

Returns the 12 Bit ADC value of the specified port on the EZ-B v4

Example: $\$x = \text{GetADC12}(\text{adc0})$

GetArraySize(variable)

Returns the size of the specified array variable.

*Note: The variable passed as a parameter must be in quotations. See the example below.

Example: $\$x = \text{GetArraySize}(\text{"$myArray"})$

GetAsByte(value)

Returns the byte of the integer or first character of string.

If you pass 75 as the value, you will get the ASCII value of the letter K

Example: $\$value = \text{GetAsByte}(\text{"H"})$

Example: $\$value = \text{GetAsByte}(\$x)$

GetBit(value, bit)

Returns the value of the specified bit. The LSB is bit 0, and MSB is bit 7.

Example: $\$bit = \text{GetBit}(255, 1)$

GetByte(value)

Returns the ASCII Ordinal value of a byte or byte array.

Technically, this function returns a number and not specifically a byte.

The number of bytes in the variable will determine the size of the integer returned. If one byte is passed, an 8 bit number is returned. If two bytes are passed, a 16 bit number is returned. If 4 bytes are passed, a 32 bit number is returned. If 6 bytes are passed, a 64 bit number is returned.

If the variable contains 0x37, this function will return an integer value of 53. Use this function to convert data read by i2c into ordinal values.

Example: $\$value = \text{GetByte}(\text{"H"})$

Example: $\$value = \text{GetByte}(\$x)$

GetByteAt(value, index)

Returns the ASCII Ordinal value of the byte at the specified location within the array.

If the byte is 0x05, this function will return an integer value of 53. Use this function to convert data read by i2c into ordinal values.

The Index is zero based, which means 0 (zero) is the first character.

Example: $\$value = \text{GetByteAt}(\text{"Hello World"}, 2)$

Example: $\$value = \text{GetByteAt}(\$x, 2)$

GetCharAt(value, index)

There are two methods to obtain a character within a string. This method, which is GetCharAt(), or using the [] square

**Halt()**

Exit the current running script.

Example: *Halt()*

HTTPGet(url)

Send an HTTP Get command to the provided address and return the contents

Example: *HTTPGet("http://192.168.0.10/decoder_control.cgi?command=35&onestep=5&user=admin&pwd=admin")*

Example: *[i]\$temp = HTTPGet("http://192.168.0.15/GetTemperature.cgi ")*



If (Value Condition Value)

IF condition can support multiple comparisons and functions.

Condition tree must be closed with an ENDIF

See the Functions section of this document.

Condition can be =, , =, !=, AND, OR

Example:

```
If (GetDigital(D0) = 1)
```

```
Print("One")
```

```
EndIf
```

Example:

```
If ($Day = 2 AND $Hour = 3)
```

```
Print("Hello")
```

```
EndIf
```

Example:

```
If (GetServo(D5) >20 OR ($x >= 3 and $y, =, !=, AND, OR
```

Example:

```
If (GetServo(D0) = 1)
```

```
Print("One")
```

```
ElseIf (GetServo(D0) = 2)
```

```
Print("Two")
```

```
Else
```

```
Print("Something Else")
```

```
EndIf
```

IndexOf(string1, string2)

Returns the index within string1 of the first occurrence of string2.

This search is case insensitive.

Example: $\$value = \text{IndexOf}(\text{"Cat In The Hat"}, \text{"In"})$

IsConnected(boardIndex)

Returns TRUE or FALSE if the specified EZ-B board index is connected

Example: $\$status = \text{IsConnected}(0)$

IsNumeric(value)

**Land()**

Tell your flying drone to land

Example: `Land()`

Left([speed], [milliSeconds])

Using a Movement Panel Control, this will turn your robot left.

You will require at least one movement panel to be configured within the project. This function will control that movement panel.

Optionally, you can specify the speed and/or number of milliseconds to turn.

Speed is a number between 0 and 255

Example #1: `Left()`

Example #2: `Left(200)`

Example #2: `Left(200, 5000)`

Length(value)

Returns the length of the specified variable or string in characters/bytes.

Example: `$len = Length("This string is 33 characters long")`

LoadProject(filename)

Loads the specified project and replaces the existing project.

Established connections will be maintained. OnConnect scripts within the Connection Control will be executed if a connection is already established.

If no path is specified, this command searches for the file in the default My Documents\EZ-Builder folder.

If no extension is provided, the .ezb default extension is assumed.

For obvious reasons, no further commands following LoadProject() are executed.

Example: `LoadProject("MyTest")`

Example: `LoadProject("MyTest.ezb")`

Example: `LoadProject("C:\Temp\MyTest.ezb")`

**Max(value1, value2)**

Returns the highest of the two values specified.

Example: *\$highest = Min(3, 5)*

Min(value1, value2)

Returns the lowest of the two values specified.

Example: *\$lowest = Min(3, 5)*

Move (servoPort, forward/stop/reverse)

Set a modified servo to move

Example: *Move(D14, "forward")*

Movement_Wait (forward/reverse/stop/left/right)

Wait until a movement from the movement panel is specified.

Even if another script executes a movement, this will execute

Example: *Movement_Wait("FORWARD")*

MP3TriggerNext(digitalPort, baud)

Plays the next MP3 track from the MP3 Trigger Shield

Example: *MP3TriggerPlayNext(d0, 38400)*

MP3TriggerPrev(digitalPort, baud)

Plays the previous MP3 track from the MP3 Trigger Shield

Example: *MP3TriggerPrevious(d0, 38400)*

MP3TriggerPlayRandomTrack(digitalPort, baud, lowestTrackNum, highestTrackNum)

Plays a random MP3 track from the MP3 Trigger Shield between the supplied track numbers

Example: *MP3TriggerPlayRandomTrack(d0, 38400, 1, 10)*

MP3TriggerStop(digitalPort, baud)

Stops the current MP3 track from the MP3 Trigger Shield

Starts the mp3 file if not playing

Example: *MP3TriggerStop(d0, 38400)*

MP3TriggerPlayTrack(digitalPort, baud, trackNumber, [pause time])

Plays the specified MP3 track from the MP3 Trigger Shield

Optionally, the Pause Time value can be used which disables the Speech Recognition control for the specified number of milliseconds.

Example: *MP3TriggerPlayTrack(d0, 38400, 1)*

Example: *MP3TriggerPlayTrack(d0, 38400, 1, 3000)*

MP3TriggerVolume(digitalPort, baud, volume)

**Pause()**

Pauses the script at the location where it is called. To resume the script, another script must call the ControlCommand's Resume function. Otherwise, the script can be stopped by pressing the Stop button on the control.

Example: *Pause()*

Ping_Wait (triggerPort, echoPort, higher/lower/equals, distance)

Wait until the Ping Sensor distance is higher or lower than specified distance value.

Trigger and Echo are Digital Ports

Example: *Ping_Wait(D3, D4, HIGHER, 50)*

PlayAudio(filename)

Plays the specified audio file to the default audio device

File formats can be MP3 or WAV

Example: *PlayAudio("c:\temp\my Audio.mp3")*

Power(value, power)

Returns the math Power() function

First parameter is the input value

The second parameter is the power

Example: *\$x = Power(2, 4)*

Print(txt)

Outputs the specified text to the debug console

Example: *Print("This is some text")*

Example: *Print("Today is \$Day")*

Example: *Print("\$pi rounded is Round(\$pi, 2)")*

PrintHex(txt)

Outputs the hex values of the specified variable to the debug console

Example: *PrintHex(\$myVariable)*

PullVar(Namespace, Cell)

Retrieve the value from the EZ-Cloud.

Example: *\$x = PullVar("DJ Sures", "test")*

PushVar(Namespace, Cell, Value)

Send the value to the EZ-Cloud.

Example: *PushVar("DJ Sures", "test", "I am testing")*

PWM (digitalPort, speed)

Set the PWM (Pulse Width Modulation) to the desired duty percentage cycle

This simulates voltage on the specified pin (Between 0 and 5v)

PWM Value is between 0 and 100

**Release (servoPort)**

Release a servo from holding its position

Example: *Release(D14)*

ReleaseAll ([boardIndex])

Release all servos from holding their position

BoardIndex is optional, and specified the EZ-B board to use

Example: *ReleaseAll()*

REPEAT

Repeats the code between REPEAT and ENDREPEAT as many times as specified. Assigns the number of times to the specified variable.

Example:

```
REPEAT($x, 0, 5, 1)
```

```
Print("x=" + $x)
```

```
ENDREPEAT
```

REPEATUNTIL

Repeats the code between REPEATUNTIL and ENDREPEATUNTIL until the specified condition is TRUE.

Example:

```
REPEATUNTIL($second = 30)
```

```
Print("Second=" + $second)
```

```
Sleep(500)
```

```
ENDREPEAT
```

REPEATWHILE

Repeats the code between REPEATWHILE and ENDREPEATWHILE until the specified condition is FALSE.

Example:

```
REPEATUNTIL($second > 50)
```

```
Print("Second=" + $second)
```

```
Sleep(500)
```

```
ENDREPEAT
```

Return()

Return from a Goto()



Say(text to speech)

Speaks the text that is specified within the brackets out of the PC Sound Card in the background. This command does not block, the script will continue to execute.

Example: *Say("Hello, I am a robot")*

SayWait(text to speech)

Speaks the text that is specified within the brackets out of the PC Sound Card and blocks until done speaking.

Example: *SayWait("Hello, I am a robot")*

SayEZB(text to speech)

Speaks the text that is specified within the brackets out of the EZ-B v4 speaker in the background. This command does not block, the script will continue to execute.

Example: *SayEZB("Hello, I am a robot")*

SendSerial(digitalPort, baudRate, data, ...)

Send a series of data over the specified port and baud rate

Data can be Hex (0x09), string ("string"), or decimal (188)

Example: *SendSerial(d0, 9600, 0x00, 0x04, 0x05)*

Example: *SendSerial(d0, 9600, 244, 200, "a")*

Example: *SendSerial(d0, 9600, "This is text")*

Example: *SendSerial(d0, 9600, "Hello " + \$name)*

SendUDP(hostname, port, data, ...)

Send a series of UDP data over the specified port to the hostname

Data can be Hex (0x09), string ("string"), or decimal (188)

Example: *SendUDP("192.168.0.1", 21, "Hello World")*

Example: *SendUDP("192.168.0.1", 21, 0x20, 0x21, 0x22, 0x30)*

Example: *SendUDP("192.168.0.1", 21, 0x20, 0x21, 0x22, "Hello")*

Example: *SendUDP("192.168.0.1", 21, 0x20, 0x21, 0x22, 0x30, \$x)*

SetBits(bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0)

Returns the value of each bit of a byte. Returns a decimal number byte of the bits. This is a useful function for i2c communication because many of the i2c devices use bits.

Example: *\$val = SetBits(1, 0, 0, 0, 0, 0, 0, 0)*

Example: *\$val = SetBits(true, false, false, false, false, false, false, false)*

SetSpeed(speed, [speedRight])

Sets the global Movement Speed value

If you specify only one parameter, the speed of both the left and right wheel will be modified

If you specify two parameters, the first parameter is the speed of the Left wheel and second parameter is the speed of the Right wheel.

The speed can be viewed in the Script Variable Viewer

Speed is between 0 (slow) and 255 (fast)

Example: *SetSpeed(50)*

Example: *SetSpeed(50, 100)*

**TakeOff()**

Tell your flying drone to take off

Example: *TakeOff()*

ToBinaryString(value)

Displays the specified value in its binary representation

Example: *\$str = ToBinaryString(254)*

ToggleDigital (digitalPort)

Toggle the digital port

Example: *ToggleDigital(D2)*

ToHex(value)

Converts the integer parameter into a readable hex value string. This will convert the integer 56 into the string "0x39". Great for debugging byte data received from i2c interface.

Example: *\$hex = ToString(\$hexData)*

ToString(value)

Converts the parameter into a string by stripping unreadable characters and terminates the end of a string with the first occurrence of a 0x00.

Example: *\$string = ToString(\$hexData)*

Tweet(message)

Send a Twitter message using the configured Twitter account. Configure your Twitter account under File->Twitter Settings.

Example: *Tweet("I Love EZ-Robot!")*

You may also use the ControlCommand to Tweet images with text from a Camera Control.

Example: *[i]ControlCommand("Camera", CameraTweet, "Our New Image")*

**UARTAvailable(boardIndex, port)**

Receive the count of bytes available in the Peripheral UART Receive Buffer of the EZ-B v4. The UART receive buffers on the EZ-B v4 are 5,000 bytes.

The Board Index is the EZ-B index starting at 0.

The port can be 0, 1 or 2.

Look near at the UART Port section lower in this document for the EZ-B Pin's associated with each UART Port.

Example: *UARTAvailable(0, 0)*

UARTInit(boardIndex, port, baudRate)

Initialize the Peripheral UART on the EZ-B v4 with the specified baud rate. The UART will stay initialized until the EZ-B v4 is power cycled, and therefore this command only needs to be called once.

The Board Index is the EZ-B index starting at 0.

The port can be 0, 1 or 2.

The baud rate can be between 1 and 3750000 bps. The UART receive buffers on the EZ-B v4 are 5,000 bytes.

Look near at the UART Port section lower in this document for the EZ-B Pin's associated with each UART Port.

Example: *UARTInit(0, 0, 9600)*

UARTRead(boardIndex, port, numBytes)

Receive ASCII bytes from the Peripheral UART Receive Buffer of the EZ-B v4. The UART receive buffers on the EZ-B v4 are 5,000 bytes.

To know how many bytes are available, use the UARTAvailable() function.

The Board Index is the EZ-B index starting at 0.

The port can be 0, 1 or 2.

Look near at the UART Port section lower in this document for the EZ-B Pin's associated with each UART Port.

Example: *UARTRead(0, 0, 10)*

Example: *UARTRead(0, 0, UARTAvailable(0, 1))*

UARTReadAvailable(boardIndex, port)

Receive all available ASCII bytes from the Peripheral UART Receive Buffer of the EZ-B v4. The UART receive buffers on the EZ-B v4 are 5,000 bytes.

The Board Index is the EZ-B index starting at 0.

The port can be 0, 1 or 2.

Look near at the UART Port section lower in this document for the EZ-B Pin's associated with each UART Port.

Example: *UARTReadAvailable(0, 0)*

Example: *UARTReadAvailable(0, 0)*

UARTReadBinary(boardIndex, port, numBytes, variable)

Receive binary bytes from the Peripheral UART Receive Buffer of the EZ-B v4 into the variable as an array. The UART receive buffers on the EZ-B v4 are 5,000 bytes.

To know how many bytes are available, use the UARTAvailable() function.

The Board Index is the EZ-B index starting at 0.

The port can be 0, 1 or 2.

Look near at the UART Port section lower in this document for the EZ-B Pin's associated with each UART Port.

Example: *UARTReadBinary(0, 0, 10, \$variable)*

Example: *UARTReadBinary(0, 0, UARTAvailable(0, 1), \$variable)*

UARTWrite(boardIndex, port, data)



References

Virtual Servo Ports

The Virtual Servo Ports do not actually control any physical hardware. You will find the Virtual Servo Ports in any control that uses a servo. These can be used in exchange of using variables for storing servo positions, or using servo controls.

V0
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19

Variable Types

Variables are global throughout the entire EZ-Builder environment. The Variable Watcher Control allows you to watch variable values in real-time. Variable types are dynamic by assignment, meaning there is no specific type definition between String, Integer and Floating point.

Examples:

String: \$str = "This is a string"
String Concat: \$str = "fu" + \$bar
Integer: \$number = 6
Floating Point: \$dec = 3.14
Byte: \$byte = 0x52
Boolean: \$bool = true
Result Condition: \$result = (\$x > \$y)
Increment numeric: \$number++
Decrement numeric: \$number--
Binary Shift Left: \$x = 123 > 1

Variable Constants/Reserved Words

These variables are read-only reserved words and cannot be assigned.

\$direction
\$date

**WaitFor(expression, [timeout MS])**

Pauses the execution of a script until the specified expression is true.

Optionally, the timeout parameter will stop waiting after the specified number of milliseconds.

Example: *WaitFor(\$AutoPositionStatus = 0)*

Example: *WaitFor(\$value1 = \$value2)*

Example: *WaitFor(\$value1 = \$value2, 1000)*

WaitForChange(value, [timeout MS])

Pauses the execution of a script until the specified value has changed.

Optionally, the timeout parameter will stop waiting after the specified number of milliseconds.

Example: *WaitForChange(\$x)*

Example: *WaitForChange(GetServo(d0))*

Example: *WaitForChange(GetDigital(d0))*

Example: *WaitForChange(GetDigital(d0), 1000)*

WaitForServoMove(servoPort, [timeout MS])

Waits for the specified servo to move.

Unlike *Servo_Wait*, this function does not wait for a specific value. It simply returns once the servo has moved to a new position.

Optionally, the timeout parameter will stop waiting after the specified number of milliseconds.

Example: *WaitForServoMove(d0)*

Example: *WaitForServoMove(d0, 1000)*

WaitForSpeech(timeOut Seconds, phrases)

Pauses and waits for one of the specified phrases to be spoken.

Returns the phrase that was spoken.

Will return "timeout" if no word is detected in the specified timeout length.

Example: *WaitForSpeech(30, "Yes", "No")*

Example: *\$value = WaitForSpeech(30, "Yes", "No")*

WaitUntilTime(hour, minute)

Waits until the specified time. The script will stop at this command and not continue until the specified time. The time is declared in 24 hour format.

Example: *WaitUntilTime(17, 30)*



Commented Text

Comment a line of code

Example: # *This is a comment. This code will not run*

:Label

Defines a label for a GOTO() command

Example: *:My_Label*

NOTE:



The script commands in this manual are subject to change such as new script commands being added, syntax changes, and removal of obsolete commands. This copy of the EZ-Script manual will be edited when changes become available, and will be made when possible.

Happy building (and scripting). :)

Manual contents correct at time of writing,

(16th October 2015).