SYNTHIAM

synthiam.com

The ARC 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 ARC. 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.

Last Updated: 10/16/2015

uEZ-Script Functions:[/b][/u]

bSleep (milliseconds)[/b] Pauses for specified milliseconds Example sleeps for 1 second: *Sleep(1000)*

bSleepRandom (lowMilliSec, highMilliSec)[/b] Pauses for a random millisecond delay between the 2 provided values Example: *SleepRandom(1000, 5000)*

bServo (servoPort, position)[/b] Move servo to the specified position Servo position is between 1 and 180 Example: *Servo(D14, 25)*

bSetServoMin (servoPort, position)[/b] Set the minimum limit that this servo can ever move to Servo position is between 1 and 180 Example: *SetServoMin(D14, 40)*

bSetServoMax (servoPort, position)[/b] Set the maximum limit that this servo can ever move to Servo position is between 1 and 180 Example: *SetServoMax(D14, 100)*

bPWM (digitalPort, speed)[/b] 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)*

bGetPWM (digitalPort)[/b] Gets the PWM (Pulse Width Modulation) of specified port PWM is between 0 and 100 Example: *$x = GetPWM(D14)*

bPWMRandom (digitalPort, lowSpeed, highSpeed)[/b] 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)*

bServoSpeed (servoPort, speed)[/b] Set the speed of servo or PWM. This is the speed to move between positions. The servo speed is a number between 0 (fastest) and 10 (slowest) b*Note:[/b] To initialize the ServoSpeed() at first use, set a Servo() position before using the ServoSpeed() command. If there is no previous position (such as during power-on), the software assumes the position is 0 and will cause issues with your robot. b*Note:[/b] Once the ServoSpeed() has been initialized the first time, specify the ServoSpeed() before specifying the Servo() position. Example: *ServoSpeed(D14, 25)*

bServoSpeedRandom (servoPort, lowSpeed, highSpeed)[/b] Set the servo speed or PWM to a random value The servo speed is a number between 0 (fastest) and 10 (slowest) b*Note:[/b] To initialize the ServoSpeed() at first use, set a Servo() position before using the ServoSpeed() command. If there is no previous position (such as during power-on), the software assumes the position is 0 and will cause issues with your robot. b*Note:[/b] Once the ServoSpeed() has been initialized the first time, specify the ServoSpeed() before specifying the Servo() position. Example: *ServoSpeedRandom(D14, 10, 20)*

bServoUp (servoPort, count)[/b] Increment the servo position value by specified count Servo position is between 1 and 180 Example: *ServoUp(D14, 1)*

bServoDown (servoPort, count)[/b] Decrement the servo position value by specified count Servo position is between 1 and 180 Example: *ServoDown(D14, 1)*

bServoRandom (servoPort, lowPosition, highPosition)[/b] Move the servo to a random position between low and high Servo position is between 1 and 180 Example: *ServoRandom(D14, 10, 20)*

bRelease (servoPort)[/b] Release a servo from holding its position Example: *Release(D14)*

bReleaseAll ([boardIndex])[/b] Release all servos from holding their position BoardIndex is optional, and specified the EZ-B board to use Example: *ReleaseAll()*

bMove (servoPort, forward/stop/reverse)[/b] Set a modified servo to move Example: *Move(D14, â€œforwardâ€)*

bSet (digitalPort, on/off/true/false)[/b] Set a digital port state to either on or off Example: *Set(D2, OFF)*

bSetRandom (digitalPort)[/b] Set a digital port to a random state of either on or off Example: *SetRandom(D2)*

bToggleDigital (digitalPort)[/b] Toggle the digital port Example: *ToggleDigital(D2)*

bDigital_Wait (digitalPort, on/off/true/false, [delay ms])[/b] 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)*

bADC_Wait (adcPort, higher/lower/equals, value, [delay ms])[/b] 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)*

bADC_Wait_Between (adcPort, low, high, [delay ms])[/b] 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)*

bMovement_Wait (forward/reverse/stop/left/right)[/b] Wait until a movement from the movement panel is specified. Even if another script executes a movement, this will execute Example: *Movement_Wait(â€œFORWARDâ€)*

bServo_Wait (digitalPort, higher/lower/equals, value)[/b] Wait until the Servo Port is higher or lower than specified value. Zero can be specified as a value for a stopped servo. Example: *Servo_Wait(D5, HIGHER, 20)*

bWaitForServoMove (servoPort, [timeout MS])[/b] 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)*

bPing_Wait (triggerPort, echoPort, higher/lower/equals, distance)[/b] 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)*

bForward([speed], [milliSeconds])[/b] Using a Movement Panel Control, this will start your robot in the Forward direction. Optionally, you can provide the speed and/or number of milliseconds to move. You will require at least one movement panel to be configured within the project. This function will control that movement panel. Speed is a number between 0 and 255 Example: *Forward()* Example: *Forward(200)* Example: *Forward(255, 5000)*

bReverse([speed], [milliSeconds])[/b] Using a Movement Panel Control, this will start your robot in the Reverse direction. Optionally, you can provide the speed and/or number of milliseconds to move. You will require at least one movement panel to be configured within the project. This function will control that movement panel. Speed is a number between 0 and 255 Example: *Reverse()* Example: *Reverse(200)* Example: *Reverse(255, 5000)*

bStop()[/b] Using Movement Panel Control, this will stop your robot. You will require at least one movement panel to be configured within the project. This function will control a movement panel. Example: *Stop()*

bLeft([speed], [milliSeconds])[/b] 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)*

bRight([speed], [milliSeconds])[/b] Using a Movement Panel Control, this will turn your robot right. 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: *Right()* Example #2: *Right(200)* Example #2: *Right(200, 5000)*

bUp([milliSeconds])[/b] Using the servo port settings from a Movement Panel Control, this will raise your drone robot up Optionally, you can specify the number of milliseconds to turn. Example #1: *Up()* Example #2: *Up(1000)*

bDown([milliSeconds])[/b] 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)*

bRollRight([milliSeconds])[/b] Using the servo port settings from a Movement Panel Control, this will roll your drone robot right Optionally, you can specify the number of milliseconds to turn. Example #1: *RollRight()* Example #2: *RollRight(1000)*

bRollLeft([milliSeconds])[/b] Using the servo port settings from a Movement Panel Control, this will roll your drone robot left Optionally, you can specify the number of milliseconds to turn. Example #1: *RollLeft()* Example #2: *RollLeft(1000)*

bLand()[/b] Tell your flying drone to land Example: Land()[/i]

bTakeOff()[/b] Tell your flying drone to take off Example: *TakeOff()*

bDroneEmergency()[/b] 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.

bAX12Led(id, on/off)[/b] 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)*

bSayEZB(text to speech)[/b] 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")*

bSayEZBWait(text to speech)[/b] Speaks the text that is specified within the brackets out of the EZ-B v4 speaker and blocks until done speaking. Example: *SayEZBWait("Hello, I am a robot")*

bStopEZBAudio()[/b] Stops any audio background that is being streamed through the EZ-B v4 Example: *StopEZBAudio()*

bSay(text to speech)[/b] 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")*

bSayWait(text to speech)[/b] 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")*

bSpeakStop( )[/b] Stops speaking the current specified phrases out of the PC Sound Card. Example: *SpeakStop()*

bSpeakRSS(url, [story index])[/b] Speaks the title and text of the rss url out of the PC Sound Card. Example #1: *SpeakRSS("http://rss.cbc.ca/lineup/world.xml")* Example #2: *SpeakRSS("http://rss.cbc.ca/lineup/world.xml", 3)*

bSpeakRSSDescription(url, [story index])[/b] Speaks only the text of the rss url out of the PC Sound Card. Example #1: *SpeakRSSDescription("http://rss.cbc.ca/lineup/world.xml")* Example #2: *SpeakRSSDescription("http://rss.cbc.ca/lineup/world.xml", 3)*

bSpeakTwitter(twitterUserName, [story index])[/b] Speaks the twitter feed for the specific username out of the PC Sound Card. Example #1: *SpeakTwitter("EZ_Robot")* Example #2: *SpeakTwitter("EZ_Robot", 3)*

bSpeakVolume(value)[/b] The volume of the speech synthesizer out of the PC Sound Card. The value is between 0 and 100 Example: *SpeakVolume(30)*

bI2CClockSpeed(boardIndex, rate)[/b] Specify the clock speed of the i2c interface. The default speed is 100000, which is 100khz. Many devices support faster speeds, up to 400000 (400khz). Example: *I2CClockSpeed(0, 100000)* Example: *I2CClockSpeed(0, 400000)*

bI2CWrite(boardIndex, deviceAddress, data, ...)[/b] Send a series of ASCII data to the specified device hex address over the i2c interface. This command will Start i2c, Write Data, and Stop i2c. boardIndex is the EZ-B you wish to use (0 is first EZ-B)

Device Hex Address of i2c device must be in 0x00 format. Data can be Hex (0x09), string ("string"), or decimal (188) Example: *I2CWrite(0, 0x09, 0x02, 0x05, 0x06)* Example: *I2CWrite(0, 0x09, 244)* Example: *I2CWrite(0, 0x09, "This is text â€œ + $variable)*

bI2CWriteBinary(boardIndex, deviceAddress, variable)[/b] Send a series of binary data to the specified device hex address over the i2c interface from the provided variable array. This command will Start i2c, Write Data, and Stop i2c. boardIndex is the EZ-B you wish to use (0 is first EZ-B) Device Hex Address of i2c device must be in 0x00 format. Variable is an array with data that you wish to send Example: *I2CWriteBinary(0, 0x09, $variable)*

bI2CRead(boardIndex, 7bitDeviceAddress, bytes to expect)[/b] Returns a series of ASCII data from the specified address over the i2c interface boardIndex is the EZ-B you wish to use (0 is first EZ-B) Device Hex Address must be hexadecimal 7Bit (0x5e) You must specify the number of Bytes To Expect Example: *$Val = I2CRead(0, 0x5e, 2)*

bI2CReadBinary(boardIndex, 7bitDeviceAddress, bytes to expect, variable)[/b] Returns a series of binary data from the specified address over the i2c interface into the specified variable array boardIndex is the EZ-B you wish to use (0 is first EZ-B) Device Hex Address must be hexadecimal 7Bit (0x5e) You must specify the number of Bytes To Expect Example: *I2CReadBinary(0, 0x5e, 2, $variable)*

bSendSerial(digitalPort, baudRate, data, ...)[/b] 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, â€œaâ€)* Example: *SendSerial(d0, 9600, "This is text")* Example: *SendSerial(d0, 9600, â€œHello â€œ + $name)*

bSendUDP(hostname, port, data, ...)[/b] 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(â€œ192.168.0.1â€, 21, â€œHello Worldâ€)* Example: *SendUDP(â€œ192.168.0.1â€, 21, 0x20, 0x21, 0x22, 0x30)* Example: *SendUDP(â€œ192.168.0.1â€, 21, 0x20, 0x21, 0x22, â€œHelloâ€)* Example: *SendUDP(â€œ192.168.0.1â€, 21, 0x20, 0x21, 0x22, 0x30, $x)*

bUARTInit(boardIndex, port, baudRate)[/b] 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 Pinâ€™s associated with each UART Port. Example: *UARTInit(0, 0, 9600)*

bUARTAvailable(boardIndex, port)[/b] 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 Pinâ€™s associated with each UART Port. Example: *UARTAvailable(0, 0)*

bUARTRead(boardIndex, port, numBytes)[/b] 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 Pinâ€™s associated with each UART Port. Example: *UARTRead(0, 0, 10)* Example: *UARTRead(0, 0, UARTAvailable(0, 1))*

bUARTReadBinary(boardIndex, port, numBytes, variable)[/b] 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 Pinâ€™s associated with each UART Port. Example: *UARTReadBinary(0, 0, 10, $variable)* Example: *UARTReadBinary(0, 0, UARTAvailable(0, 1), $variable)*

bUARTReadAvailable(boardIndex, port)[/b] 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 Pinâ€™s associated with each UART Port. Example: *UARTReadAvailable(0, 0)* Example: *UARTReadAvailable(0, 0)*

bUARTWrite(boardIndex, port, data)[/b] Write ASCII data through the Peripheral UART. 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 Pinâ€™s associated with each UART Port. Example: *UARTWrite(0, 0, â€œhello worldâ€)* Example: *UARTWrite(0, 0, 0x30, 0x40, â€œhelloâ€)*

bUARTWriteBinary(boardIndex, port, variable)[/b] Write binary variable array data through the Peripheral UART. 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 Pinâ€™s associated with each UART Port. Example: *UARTWriteBinary(0, 0, $variable)*

bWaitUntilTime(hour, minute)[/b] 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)*

bMP3TriggerPlayTrack(digitalPort, baud, trackNumber, [pause time])[/b] Plays the specified MP3 track from the MP3 Trigger Sheild 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)*

bMP3TriggerVolume(digitalPort, baud, volume)[/b] Set the volume of the mp3 trigger between 0 and 255. 0 is louded, 255 is quiet. Example: *MP3TriggerVolume(d0, 38400, 20)*

bMP3TriggerPlayRandomTrack(digitalPort, baud, lowestTrackNum, highestTrackNum)[/b] Plays a random MP3 track from the MP3 Trigger Shield between the supplied track numbers Example: *MP3TriggerPlayRandomTrack(d0, 38400, 1, 10)*

bMP3TriggerNext(digitalPort, baud)[/b] Plays the next MP3 track from the MP3 Trigger Shield Example: *MP3TriggerPlayNext(d0, 38400)*

bMP3TriggerPrev(digitalPort, baud)[/b] Plays the previous MP3 track from the MP3 Trigger Shield Example: *MP3TriggerPrevious(d0, 38400)*

bMP3TriggerStop(digitalPort, baud)[/b] Stops the current MP3 track from the MP3 Trigger Shield Starts the mp3 file if not playing Example: *MP3TriggerStop(d0, 38400)*

bControlCommand(windowName, ControlCommandParameter, [values])[/b] Sends a command to the window by its name. View all available ControlCommand() parameters for each control when editing a script by pressing the Cheat Sheet tab. The Cheat Sheet tab is located to the right of the script editor in EZ-Builder. Pressing Cheat Sheet tab will display a list of all available parameters for each control. Simply click on the parameter and it will insert into your code. Some commands require an additional parameter. When editing EZ-Script, check the Cheat Sheet tab to view all available commands for the controls within your project. This command has a shorthand alias which is â€œCCâ€ (See examples below). Example: *ControlCommand("ADC Graph", pauseOn)* Example: *ControlCommand("SoundBoard", Track_3)* Example: *ControlCommand("Camera", CameraTweet, "This is an Image Description")* Example: *ControlCommand("Script Manager", ScriptStart, "MyScript")* Example: *ControlCommand("Speech Recognition", PauseMS, 3000)* Example: *cc(â€œAuto Positionâ€, AutoPositionAction, â€œAction Nameâ€)* Example: *cc(â€œAuto Positionâ€, AutoPositionFrame, â€œFrame Nameâ€)* Example: *cc(â€œAuto Positionâ€, AutoPositionFrame, â€œFrame Nameâ€, 50, 3)* Example: *cc(â€œSpeech Recognitionâ€, PauseMS, 2000)*

bGetControlValue(windowName, ControlCommandValues)[/b] Gets a value from the window by its name. Look further down in this document for available control command parameters under the ControlCommand Values section. Example: [i]$x = GetControlValue("ADC Graph", â€œpauseâ€)

b# Commented Text[/b] Comment a line of code Example: *# This is a comment. This code will not run*

Defines a label for a GOTO() command Example: *:My_Label*

bGoto(label)[/b] Goto a specific :Label location Example: *Goto(My_Label)*

bReturn()[/b] Return from a Goto() If you jump to a position of code with a Goto(), the Return statement will allow you to return back to that piece of code following the last Goto() statement. If you attempt to Return() with an empty stack, nothing will happen. The script will ignore the Return() statement. Example: *Return()*

bIf (Value Condition Value)[/b] 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(â€œOneâ€) EndIf* Example: *If ($Day = 2 AND $Hour = 3) Print(â€œHelloâ€) EndIf* Example: *If (GetServo(D5) >20 OR ($x >= 3 and $y , =, !=, AND, OR* Example: *If (GetServo(D0) = 1) Print(â€œOneâ€) ElseIf (GetServo(D0) = 2) Print(â€œTwoâ€) Else Print(â€œSomething Elseâ€) EndIf*

bElse[/b] Condition tree must be closed with an ENDIF Example: *If (GetDigital(D0) = 1) Print(â€œYesâ€) Else Print(â€œNoâ€) EndIf*

bREPEAT[/b] 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(â€œx=â€ + $x) ENDREPEAT*

bREPEATUNTIL[/b] Repeats the code between REPEATUNTIL and ENDREPEATUNTIL until the specified condition is TRUE. Example: *REPEATUNTIL($second = 30) Print(â€œSecond=â€ + $second) Sleep(500) ENDREPEAT*

bREPEATWHILE[/b] Repeats the code between REPEATWHILE and ENDREPEATWHILE until the specified condition is FALSE. Example: *REPEATUNTIL($second > 50) Print(â€œSecond=â€ + $second) Sleep(500) ENDREPEAT*

bGetServo(Port)[/b] Returns the last specified Servo Position value of the servo port. Servo position is between 1 and 180 Example: *$x = GetServo(d0)*

bGetServoSpeed(Port)[/b] Returns the Servo Speed value of the specified port Speed is between 0 and 10 Example: *$x = GetServoSpeed(d0)*

bSetSpeed(speed, [speedRight])[/b] 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)*

bSetSpeedLeft(speed)[/b] Sets the global Movement Speed value of the Left wheel Speed is between 0 (slow) and 255 (fast) The speed can be viewed in the Script Variable Viewer Example: *SetSpeedLeft(50)*

bSetSpeedRight(speed)[/b] Sets the global Movement Speed value of the Right wheel Speed is between 0 (slow) and 255 (fast) The speed can be viewed in the Script Variable Viewer Example: *SetSpeedRight(50)*

bGetSpeed( )[/b] Returns the global Movement Speed value Speed is between 0 and 255 The speed can be viewed in the Script Variable Viewer Example: *$x = GetSpeed()*

bGetSpeedLeft( )[/b] Returns the global Movement Speed value of Left wheel Speed is between 0 and 255 The speed can be viewed in the Script Variable Viewer Example: *$x = GetSpeedLeft()*

bGetSpeedRight( )[/b] Returns the global Movement Speed value of Right wheel Speed is between 0 and 255 The speed can be viewed in the Script Variable Viewer Example: *$x = GetSpeedRight()*

bGetCPUTemp()[/b] Returns the CPU Temperature of the EZ-B v4 Example: *$x = GetCPUTemp()*

bGetVoltage()[/b] Returns the EZ-B v4 Battery Voltage Example: *$x = GetVoltage()*

bGetADC(Port)[/b] Returns the 8 Bit ADC value of the specified port Example: *$x = GetADC(adc0)*

bGetADC12(Port)[/b] Returns the 12 Bit ADC value of the specified port on the EZ-B v4 Example: *$x = GetADC12(adc0)*

bGetRandom(Min, Max)[/b] Returns a random number between Min and Max

Example: *$x = GetRandom(10, 50)*

bGetRandomUnique(Min, Max)[/b] Returns a random number between Min and Max This function attempts to make the number unique from the last time it was called Example: *$x = GetRandomUnique(10, 50)*

bGetDigital(Port)[/b] Returns the Digital value of the specified port as a 0 or 1 Example: *$x = GetDigital(d0)*

bASin(value)[/b] Returns the math ASin() function (also called ArcSin) Example: *$x = ASin(27)*

bACos(value)[/b] Returns the math ACos() function (also called ArcCos) Example: *$x = Acos(27)*

bSqrt(value)[/b] Returns the math Square Root function Example: *$x = Sqrt(9)*

bPower(value, power)[/b] Returns the math Power() function First parameter is the input value The second parameter is the power Example: *$x = Power(2, 4)*

bSin(value)[/b] Returns the math SIN() function Example: *$x = Sin(27)*

bCos(value)[/b] Returns the math COS() function Example: *$x = Cos(27)*

bAbs(value)[/b] Returns the absolute value of a number Converts a negative into a positive number Example: *$x = Abs(-22)*

bRound(value, decimals)[/b] Returns the math Round() of a number Returns the number rounded to the specified decimal places Example: *$x = Round($pi, 2)* Example: *$x = Round(9.3848291, 1)*

bGetPing(trigger port, echo port)[/b] Return the Ping HC-SR04 value of the specified port Example: *$x = GetPing(d0, d1)*

bGPSStop(latitude, longitude, resolution)[/b] Uses the attached GPS control and stops the movement panel when the coordinates are within the specified resolution for the latitude and longitude. Example: *GPSStop(54.01438, -110.4931, 0.0005)*

bRoboSapien(RoboSapienCommand)[/b] Send a command to a RoboSapien connected on port D1 on EZ-B 0. Find the available RoboSapien commands further down in this document. Example: *RoboSapien(â€œWalkForwardâ€)* Example: *RoboSapien(â€œLeftArmUpâ€)*

bRoboQuad(RoboQuadCommand)[/b] Send a command to a RoboQuad connected on port D1 on EZ-B 0. Find the available RoboQuad commands further down in this document. Example: *RoboQuad(â€œWalk_Forwardâ€)* Example: *RoboQuad(â€œLeft_Turn_Rollâ€)*

bTweet(message)[/b] 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")*

bHTTPGet(url)[/b] 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 ")*

bRoomba(cmd)[/b] Execute the specified command on a connected Roomba Vacuum on Port D0 and EZ-B 0. Look for the available Roomba commands further below in this document. You may also add the Roomba Movement Panel for graphical controls. Example: *Roomba(â€œinitâ€)* Example: *Roomba(â€œSideBrushOnâ€)*

bSoundNote(note, lengthMS, [signal type])[/b] Plays the specified audio note out of the EZ-B v4 speaker for the specified number of milliseconds. Optionally, you may provide a signal type as well. The valid signal types are Sine, Square, Triangle, Pulse, Sawtooth, WhiteNoise, GaussNoise, DigitalNoise. Example: *SoundNote(â€œC2â€, 1000)* Example: *SoundNote(â€œC2â€, 1000, â€œSquareâ€)*

bHalt()[/b] Exit the current running script. Example: *Halt()*

bPrint(txt)[/b] Outputs the specified text to the debug console Example: *Print(â€œThis is some textâ€)* Example: *Print(â€œToday is $Dayâ€)* Example: *Print(â€œ$pi rounded is Round($pi, 2)â€)*

bPrintHex(txt)[/b] Outputs the hex values of the specified variable to the debug console Example: *PrintHex($myVariable)*

bExec(EXE/Bat File, [parameters])[/b] Executes a windows application or batch file. The second parameter is a list of optional parameters Example: *Exec(â€œC:\Windows\Notepad.exeâ€)* Example: *Exec(â€œC:\Windows\Notepad.exeâ€, â€œC:\MyFile.txtâ€)*

bBrowser(url)[/b] Launches the default web browser with the specified URL. Example: *Browser(â€œhttp://www.google.comâ€)*

bFileDelete(filename)[/b] Deletes a file on your computer Example: *FileDelete(â€œc:\temp\mylog.txtâ€)*

bFileWrite(filename, contents)[/b] Appends text to the specified file. This does not append a new line. Example: *FileWrite(â€œc:\temp\mylog.txtâ€, â€œMy Variable: â€œ + $x)* Example: *FileWrite(â€œc:\temp\mylog.txtâ€, â€œServo Position: GetServo(d2)â€)*

bFileWriteLine(filename, contents)[/b] Appends text as a new line to the specified file. Example: *FileWriteLine(â€œc:\temp\mylog.txtâ€, â€œMy Variable: â€œ + $x)* Example: *FileWriteLine(â€œc:\temp\mylog.txtâ€, â€œServo Position: GetServo(d2)â€)*

bFileReadClose(filename)[/b] 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(â€œc:\temp\mylog.txtâ€)*

bFileReadReset(filename)[/b] 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(â€œc:\temp\mylog.txtâ€)*

bFileExists(filename)[/b] Returns a 1 or 0 if the specified file exists. Example: *$fileExists = FileExists(â€œc:\temp\mylog.txtâ€)*

bFileReadEnd(filename)[/b] Returns a 1 or 0 if the file has reached the end. Example: *$fileEnd = FileReadEnd(â€œc:\temp\mylog.txtâ€)*

bFileReadChar(filename)[/b] Returns the next character in the specified file Example: *$char = FileReadChar(â€œc:\temp\mylog.txtâ€)*

bFileReadLine(filename)[/b] Returns the next line of the specified filename. Example: *$line = FileReadLine(â€œc:\temp\mylog.txtâ€)*

bFileReadAll(filename)[/b] Returns the entire contents of the specified file. Example: *$contents = FileReadAll(â€œc:\temp\mylog.txtâ€)*

bFileReadLineRandom(filename)[/b] Returns a random line within the specified file Example: *$randomLine = FileReadLineRandom(â€œc:\temp\mylog.txtâ€)*

bSplit(text, splitChar, index)[/b] Splits a line of text by the specified SplitChar into an array and returns the Index. The Index is zero based, which means 0 (zero) is the first item within the array. Example: *$contents = Split(â€œOne, Two, Threeâ€, â€œ,â€, 1)* Example: *$contents = Split(â€œOne - Two - Threeâ€, â€œ-â€, 2)*

bWaitForChange(value, [timeout MS])[/b] 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)*

bWaitFor(expression, [timeout MS])[/b] 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)*

bLength(value)[/b] Returns the length of the specified variable or string in characters/bytes. Example: *$len = Length(â€œThis string is 33 characters longâ€)*

bGetCharAt(value, index)[/b] There are two methods to obtain a character within a string. This method, which is GetCharAt(), or using the [] square brackets. See the example below. Returns the character at the specified index. If the character at the specified position is outside of readable ASCII, you will want to use GetByteAt() or GetByte() functions instead. The Index is zero based, which means 0 (zero) is the first character. Example: *$char = GetCharAt(â€œHello Worldâ€, 2)* Example: *$char = GetCharAt($x, 2)* Example: *$byte = $x[3]* Example: *$byte = $x[$y]*

bGetByteAt(value, index)[/b] 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: *$value = GetByteAt(â€œHello Worldâ€, 2)* Example: *$value = GetByteAt($x, 2)*

bGetByte(value)[/b] 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: *$value = GetByte(â€œHâ€)* Example: *$value = GetByte($x)*

bGetAsByte(value)[/b] 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: *$value = GetAsByte(â€œHâ€)* Example: *$value = GetAsByte($x)*

bClearVariables( )[/b] 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()*

bAppendArray(variable, value)[/b] 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, â€œbananaâ€)*

bDefineArray(variable, size, [default value])[/b] 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)*

bFillArray(variable, default value)[/b] Fill an existing array with the default value. Example: *FillArray($myArray, 88)* Example: *FillArray($myArray, â€œdefault valueâ€)*

bGetArraySize(variable)[/b] Returns the size of the specified array variable. *Note: The variable passed as a parameter must be in quotations. See the example below. Example: *$x = GetArraySize(â€œ$myArrayâ€)*

bDumpVariables( )[/b] 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()*

bIsNumeric(value)[/b] Returns True or False if the specified value is numeric. Example: *$value = IsNumeric($x)*

bWaitForSpeech(timeOut Seconds, phrases)[/b] Pauses and waits for one of the specified phrases to be spoken. Returns the phrase that was spoken. Will return â€œtimeoutâ€ if no word is detected in the specified timeout length. Example: *WaitForSpeech(30, â€œYesâ€, â€œNoâ€)* Example: *$value = WaitForSpeech(30, â€œYesâ€, â€œNoâ€)*

bContains(string1, string2)[/b] Returns TRUE or FALSE if string2 is found within string1. This search is case insensitive. Example: *$value = Contains(â€œCat In The Hatâ€, â€œCatâ€)*

bIndexOf(string1, string2)[/b] Returns the index within string1 of the first occurrence of string2. This search is case insensitive. Example: *$value = IndexOf(â€œCat In The Hatâ€, â€œInâ€)*

bSubString(string1, start, length)[/b] Returns the specified substring within string1. Example: *$value = SubString(â€œCat In The Hatâ€, 4, 2)*

bPlayAudio(filename)[/b] Plays the specified audio file to the default audio device File formats can be MP3 or WAV Example: *PlayAudio(â€œc:\temp\my Audio.mp3â€)*

bStopAudio()[/b] Stops the current audio file that is playing through the default audio device by PlayAudio() Example: *StopAudio()*

bIsConnected(boardIndex)[/b] Returns TRUE or FALSE if the specified EZ-B board index is connected Example: *$status = IsConnected(0)*

bToString(value)[/b] 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)*

bMin(value1, value2)[/b] Returns the lowest of the two values specified. Example: *$lowest = Min(3, 5)*

bMax(value1, value2)[/b] Returns the highest of the two values specified. Example: *$highest = Min(3, 5)*

bToHex(value)[/b] Converts the integer parameter into a readable hex value string. This will convert the integer 56 into the string â€œ0x39â€. Great for debugging byte data received from i2c interface. Example: *$hex = ToString($hexData)*

bSetBits(bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0)[/b] 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)*

bToBinaryString(value)[/b] Displays the specified value in its binary representation Example: *$str = ToBinaryString(254)*

bGetBit(value, bit)[/b] Returns the value of the specified bit. The LSB is bit 0, and MSB is bit 7. Example: *$bit = GetBit(255, 1)*

bPause( )[/b] Pauses the script at the location where it is called. To resume the script, another script must call the ControlCommandâ€™s Resume function. Otherwise, the script can be stopped by pressing the Stop button on the control. Example: *Pause()*

bComOpen(Port, Baud Rate)[/b] 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(â€œcom1â€, 9600)*

bComClose(Port)[/b] Close the specified serial communication port on the local PC. Example: *ComClose(â€œcom1â€)*

bComWrite(Port, String)[/b] Write the string to the specified serial communication port on the local PC. The port must be open before this command can be called. Example: *ComWrite(â€œcom1â€, â€œthis is dataâ€)* Example: *ComWrite(â€œcom1â€, $variable)*

bComWriteLine(Port, String)[/b] Write the string as a new line to the specified serial communication port on the local PC. The port must be open before this command can be called. Example: *ComWriteLine(â€œcom1â€, â€œThis is a lineâ€)* Example: *ComWriteLine(â€œcom1â€, $variable)*

bComReadLine(Port)[/b] Read all string data up to a newline 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 = ComReadLine(â€œcom1â€)*

bComReadAll(Port)[/b] 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(â€œcom1â€)*

bComAvailable (Port)[/b] 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(â€œcom1â€)*

bComRead(Port, Bytes to Read)[/b] 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(â€œcom1â€, 15)*

bComClearInput(Port)[/b] Clear the input buffer. The port must be open before this command can be called. Example: *ComClearInput(â€œcom1â€)*

bComWriteBinary(Port, Array)[/b] Write the array of data to the specified serial communication port on the local PC. The port must be open before this command can be called. Example: *ComWriteBinary(â€œcom1â€, $arrayVariable)*

bComReadBinary(Port, Bytes to Read, Array)[/b] Read specified number of byes from the input buffer of the specified serial communication port on the local PC. The data will be stored in the specified array. The array will be created automatically and sized to the incoming data. The port must be open before this command can be called. Example: *ComReadBinary(â€œcom1â€, 10, $arrayVariable)*

bPushVar(NameSpace, Cell, Value)[/b] Send the value to the EZ-Cloud. Example: *PushVar(â€œDJ Suresâ€, â€œtestâ€, â€œI am testingâ€)*

bPullVar(NameSpace, Cell)[/b] Retrieve the value from the EZ-Cloud. Example: *$x = PullVar(â€œDJ Suresâ€, â€œtestâ€)*

bShowControl(ControlName)[/b] Used for mobile devices and the Interface Builder only, this command will open the specified control into the foreground. Example: *ShowControl(â€œWii Remoteâ€)*

bCloseControl( )[/b] 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()*

bShowDesktop(desktopNumber)[/b] Shows the specified virtual desktop. The desktop number is 1, 2 or 3. Example: *ShowDesktop(1)*

bSetVolume(volume)[/b] Sets the volume of the EZ-B v4 speaker The volume can be a value between 0 (quite), 100 (loud), 200 (2x over drive) The volume can be viewed in the Script Variable Viewer Example: *SetVolume(50)* Example: *SetVolume(100)* Example: *SetVolume(150)*

bGetVolume()[/b] Gets the volume of the EZ-B v4 speaker The volume will be a value between 0 (quite), 100 (loud), 200 (2x over drive) The volume can be viewed in the

Script Variable Viewer Example: *$x = GetVolume(50)*

bSleepPC(Suspend|Hibernate, force, wake)[/b] Sends a command to the operating system to sleep or hibernate. If Force is TRUE, the computer is forced to sleep and other applications have no say in the decision. If Wake is TRUE, the computer will wake up on Wake events. Example: *SleepPC(Suspend, true, true)* Example: *SleepPC(Hibernate, true, true)*

bLoadProject(filename)[/b] 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(â€œMyTestâ€)* Example: *LoadProject(â€œMyTest.ezbâ€)* Example: *LoadProject(â€œC:\Temp\MyTest.ezbâ€)*

bCheckForUpdate()[/b] Checks the EZ-Robot server if there is a newer version of EZ-Builder available. This requires an internet connection. Example: *$x = CheckForUpdate()*

uReferences[/b][/u]

bMultiple EZ-B Boards[/b]

EZ-Builder supports multiple physical EZ-B Boards connected to your computer. You can specify the board by putting the board number in front of the port. For example: Servo(2.d0, 8) will move the D0 servo on EZ-B board #2 to position 8. If no board index is specified, the first board (zero) is assumed. If using more than one board, the first board is always responsible for movement panels.

bADC Ports[/b] 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. ADC0 ADC1 ADC2 ADC3 ADC4 ADC5 ADC6 ADC7

bVirtual Servo Ports[/b] 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

bServo/Digital Ports[/b] 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 D11 D12 D13 D14 D15 D16 D17 D18 D19 D20 D21 D22 D23

bDynamixel Servos (protocol v1) for AX/RX/EX (Port D5)[/b] Standard hobby servos use a pulse width modulation for setting the position. Optionally, more advanced users may use the higher priced Dynamixel servos for increased position resolution and strength. These servos are connected together in a chain of wiring (each servo connects to the previous servo). Connect the signal wire of the servo chain to the D5

Port of the EZ-B v4. This feature does not work on the EZ-B v3. Â· AX0 Â· AX1 Â· AX2 Â· AX3 Â· AX4 Â· AX5 Â· AX6 Â· AX7 Â· AX8 Â· AX9 Â· AX10 Â· AX11 Â· AX12 Â· AX13 Â· AX14 Â· AX15 Â· AX16 Â· AX17 Â· AX18 Â· AX19 Â· AX20 Â· AX21 Â· AX22 Â· AX23 Â· AX24 Â· AX25 Â· AX26 Â· AX27 Â· AX28 Â· AX29 Â· AX30 Â· AX31 Â· AX32 Â· AX33 Â· AX34 Â· AX35 Â· AX36 Â· AX37 Â· AX38 Â· AX39 Â· AX40 Â· AX41 Â· AX42 Â· AX43 Â· AX44 Â· AX45 Â· AX46 Â· AX47 Â· AX48 Â· AX49 Â· AX50

bDynamixel Servos (protocol v2) for XL-320 & Pro (Port D5)[/b] Standard hobby servos use a pulse width modulation for setting the position. Optionally, more advanced users may use the higher priced Dynamixel servos for increased position resolution and strength. These servos are connected together in a chain of wiring (each servo connects to the previous servo). Connect the signal wire of the servo chain to the D5 Port of the EZ-B v4. This feature does not work on the EZ-B v3. â€¢ AXV0 â€¢ AXV1 â€¢ AXV2 â€¢ AXV3 â€¢ AXV4 â€¢ AXV5 â€¢ AXV6 â€¢ AXV7 â€¢ AXV8 â€¢ AXV9 â€¢ AXV10 â€¢ AXV11 â€¢ AXV12 â€¢ AXV13 â€¢ AXV14 â€¢ AXV15 â€¢ AXV16 â€¢ AXV17 â€¢ AXV18 â€¢ AXV19 â€¢ AXV20 â€¢ AXV21 â€¢ AXV22 â€¢ AXV23 â€¢ AXV24 â€¢ AXV25 â€¢ AXV26 â€¢ AXV27 â€¢ AXV28 â€¢ AXV29 â€¢ AXV30 â€¢ AXV31 â€¢ AXV32 â€¢ AXV33 â€¢ AXV34 â€¢ AXV35 â€¢ AXV36 â€¢ AXV37 â€¢ AXV38 â€¢ AXV39 â€¢ AXV40 â€¢ AXV41 â€¢ AXV42 â€¢ AXV43 â€¢ AXV44 â€¢ AXV45 â€¢ AXV46 â€¢ AXV47 â€¢ AXV48 â€¢ AXV49 â€¢ AXV50

bUART Ports[/b] The UARTx ports are used connect to Serial TTL devices for both input and output. Contrary to the digital port Serial Output, these peripherals will also receive data into an input buffer as well. The input buffer of each UART is 5,000 Bytes. There are 3 UARTs, the first is the hardware labelled port, second and third are digital pins. These UARTs are controlled using the UARTInit(), UARTWrite(), UARTRead() and UARTAvailable() commands. The speed of these UARTs can be any integer between 1 and 3750000 bps. Â· UART0 TX: Expansion Connector Â· UART0 RX: Expansion Connector Â· UART1 TX: D5 Â· UART1 RX: D6 Â· UART2 TX: D18 Â· UART2 RX: D19

bDigital Port Outout Baud Rates[/b] The digital ports can output Serial, which differs from the UART. Using any digital port as a Serial Output command can be done by using the SendSerial() command. Using a digital port this way does not include an input buffer. For input buffers, use the UARTx functions. 300 1200 2400 4800 9600 19200 38400 57600 115200

bRoboQuad Commands[/b] Supported on the EZ-B v3 only, the D1 port can be used to connect to the IR wire of some WowWee robots. Due to multiple hardware revisions of the robots, these functions are not always compatible and therefore this feature has been discontinued in the EZ-B v4. Stop Walk_Forward Right_Crab_Walk Left_Crab_Walk Left_Crab_Four_Steps Right_Crab_Four_Steps Backward_Four_Steps Walk_Backward Forward_Four_Steps Rotate_Counter_Clockwise Counter_Clockwise_Four_Steps Rotate_Clockwise Clockwise_Four_Steps Head_Up Head_Down Head_Clockwise Head_Counter_Clockwise Top_Left_Shuffle Top_Right_Shuffle Bottom_Left_Shuffle Bottom_Right_Shuffle Left_Strafe Right_Strafe Left_Turn_Roll Right_Turn_Roll Burst Single_Shot Stomp_Walk Left_Legs_In Left_Legs_Out Left_Forward_Leg_In Left_Forward_Leg_Out Left_Backward_Leg_In Left_Backward_Leg_Out Right_Legs_In Right_Legs_Out Right_Forward_Leg_In Right_Forward_Leg_Out Right_Backward_Leg_In Right_Backward_Leg_Out Program Play_Program Program_Delete_Last_Step Erase_Program Scan_Left_For_Object Scan_Right_For_Object Smart_Scan

Approach_Nearest_Object Escape_Walk Toggle_Activity_Level_1 Toggle_Activity_Level_2 Toggle_Activity_Level_3 Toggle_Aggression_1 Toggle_Aggression_2 Toggle_Aggression_3 Toggle_Awareness_1 Toggle_Awareness_2 Toggle_Awareness_3 Leg_Reset Full_Reset Volume_Up Volume_Down Guard Sleep Toggle_Autonomy Toggle_Sensors Twitch Surprise Wave Dizzy Attack Roar Aware_Stance High_Stance Aggressive_Stance Dance_Demo Movement_Demo Leg_Check

bRoboSapien Commands[/b] Supported on the EZ-B v3 only, the D1 port can be used to connect to the IR wire of some WowWee robots. Due to multiple hardware revisions of the robots, these functions are not always compatible and therefore this feature has been discontinued in the EZ-B v4.

TurnRight RightArmUp RightArmOut TiltBodyRight RightArmDown RightArmIn WalkForward WalkReverse TurnLeft LeftArmUp LeftArmOut TiltBodyLeft LeftArmDown LeftArmIn Stop WakeUp Burp RightHandStrike RightHandSweep RightHandStrike2 HighFive Fart LeftHandStrike LeftHandSweep Whistle Roar

bControlCommand Values[/b] You can receive values for controls using using the GetControlValue() command. The syntax for this command can be found above in this document. Each control will accept a different collection of commands, which are listed here.

All Controls Pause

bVariable Types[/b] 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.

bExamples:[/b] String: $str = â€œThis is a stringâ€ String Concat: $str = â€œfuâ€ + $bar Integer: $number = 6 Floating Point: $dec = 3.14 Byte: $byte = 0x52 Boolan: $bool = true Result Condition: $result = ($x > $y) Increment numeric: $number++ Decrement numeric: $number- Binary Shift Left: $x = 123 > 1

bScientific Math Functions[/b] Sin() Cos() Tan() Sec() Csc() Cot() ASin() ACos() ATan() SinH() CosH() TanH() Abs() Sqrt() Ciel() Floor() Exp() Log10() Log() Max() Min() Round() E() Pi() Now() Today()

bDateTime Functions[/b] MinDate() MaxDate() MonthName() AddDays() AddMonths() AddYears() AddHours() AddMinute() AddSeconds() FmtNum() FmtDate()

bCasting Functions[/b] These functions are to cast objects from one datatype to another To Double: CDBL() To Integer: CInt() To Long: CLong() To Unsigned Integer: CUint() To Unsigned Long: CULong() To DateTime: CDateTime()

bVariable Constants/Reserved Words[/b] These variables are read-only reserved words and cannot be assigned. $direction $date $month $year $day $dayName $hour $minute $second $monthName $time $pi

bRoomba Commands[/b] The iRobot Roomba can be controlled by the EZ-B. Once the Roomba Movement Panel has been added, these commands can be parameters to the Roomba() function. Syntax use for this function can be found above in this document. Init EnableSensors DisableSensors PowerOff SpotClean Clean MaxClean

DisableAllBrushes MainBrushOn MainBrushOff SideBrushOn SideBrushOff VacuumOn VacuumOff SeekDockingStation

bRoomba/Sound Music Notes[/b] The iRobot Roomba or EZ-B v4 can be told to play audio tones. They may be simple, but it is fun - here is a list of parameters for the RoombaTone() or SoundNote() functions. The syntax for the function can be found above in this document. Here are the acceptable notesâ€¦ C1 Db1 D1 Eb1 E1 F1 Gb1 G1 Ab1 A1 Bb1 B1 C2 Db2 D2 Eb2 E2 F2 Gb2 G2 Ab2 A2 Bb2

The notes below are not supported with iRobot Roomba and only works with EZ-B v4 SoundNote() C3 Db3 D3 Eb3 E3 F3 Gb3 G3 Ab3 A3 Bb3

bAuto Start Command Line Options[/b]

EZ-Builder supports two command line options for loading a project, and/or executing a script. If you wish to have EZ-Builder auto-load from a shortcut, the command line parameters allow you to specify an auto connect also. You may use the Shortcut Creator, located under the File Menu.

Parameter 1: The path and filename of the project file you wish to load apon startup.
Parameter 2: The name of the script control you wish to execute apon startup.

"c:\Program Files\EZ-Builder\EZ-Builder.exe "C:\files\My File.ezb" "c:\Program Files\EZ-Builder\EZ-Builder.exe "C:\files\My File.ezb" "InitScript"

ACos(value) Returns the math ACos() function (also called ArcCos) Example: *$x = Acos(27)*

Abs(value) Returns the absolute value of a number Converts a negative into a positive number Example: *$x = 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, â€œbananaâ€)*

ASin(value) Returns the math ASin() function (also called ArcSin) Example: *$x = 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. ADC0 ADC1 ADC2 ADC3 ADC4 ADC5 ADC6 ADC7

Auto Start Command Line Options

EZ-Builder supports two command line options for loading a project, and/or executing a script. If you wish to have EZ-Builder auto-load from a shortcut, the command line parameters allow you to specify an auto connect also. You may use the Shortcut Creator, located under the File Menu.

Parameter 1: The path and filename of the project file you wish to load apon startup. Parameter 2: The name of the script control you wish to execute apon startup.

"c:\Program Files\EZ-Builder\EZ-Builder.exe "C:\files\My File.ezb" "c:\Program Files\EZ-Builder\EZ-Builder.exe "C:\files\My File.ezb" "InitScript"

B.)

Browser(url) Launches the default web browser with the specified URL. Example: *Browser(â€œhttp://www.google.comâ€)*

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(â€œcom1â€)*

ComClearInput(Port) Clear the input buffer. The port must be open before this command can be called. Example: *ComClearInput(â€œcom1â€)*

ComClose(Port) Close the specified serial communication port on the local PC. Example: *ComClose(â€œcom1â€)*

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(â€œcom1â€, 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(â€œcom1â€, 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(â€œcom1â€)*

ComReadBinary(Port, Bytes to Read, Array) Read specified number of byes from the input buffer of the specified serial communication port on the local PC. The data will be stored in the specified array. The array will be created automatically and sized to the incoming data. The port must be open before this command can be called. Example: *ComReadBinary(â€œcom1â€, 10, $arrayVariable)*

ComReadLine(Port) Read all string data up to a newline 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 = ComReadLine(â€œcom1â€)*

ComWrite(Port, String) Write the string to the specified serial communication port on the local PC. The port must be open before this command can be called. Example: *ComWrite(â€œcom1â€, â€œthis is dataâ€)* Example: *ComWrite(â€œcom1â€, $variable)*

ComWriteBinary(Port, Array) Write the array of data to the specified serial communication port on the local PC. The port must be open before this command can be called. Example: *ComWriteBinary(â€œcom1â€, $arrayVariable)*

ComWriteLine(Port, String) Write the string as a new line to the specified serial communication port on the local PC. The port must be open before this command can be called. Example: *ComWriteLine(â€œcom1â€, â€œThis is a lineâ€)* Example: *ComWriteLine(â€œcom1â€, $variable)*

Contains(string1, string2) Returns TRUE or FALSE if string2 is found within string1. This search is case insensitive. Example: *$value = Contains(â€œCat In The Hatâ€, â€œCatâ€)*

ControlCommand(windowName, ControlCommandParameter, [values]) Sends a command to the window by its name. View all available ControlCommand() parameters for each control when editing a script by pressing the Cheat Sheet tab. The Cheat Sheet tab is located to the right of the script editor in EZ-Builder. Pressing Cheat Sheet tab will display a list of all available parameters for each control. Simply click on the parameter and it will insert into your code. Some commands require an additional parameter. When editing EZ-Script, check the Cheat Sheet tab to view all available commands for the controls within your project. This command has a shorthand alias which is â€œCCâ€ (See examples below). Example: *ControlCommand("ADC Graph", pauseOn)* Example: *ControlCommand("SoundBoard", Track_3)* Example: *ControlCommand("Camera", CameraTweet, "This is an Image Description")* Example: *ControlCommand("Script Manager", ScriptStart, "MyScript")* Example: *ControlCommand("Speech Recognition", PauseMS, 3000)* Example: *cc(â€œAuto Positionâ€, AutoPositionAction, â€œAction Nameâ€)* Example: *cc(â€œAuto Positionâ€, AutoPositionFrame, â€œFrame Nameâ€)* Example: *cc(â€œAuto Positionâ€, AutoPositionFrame, â€œFrame Nameâ€, 50, 3)* Example: *cc(â€œSpeech Recognitionâ€, PauseMS, 2000)*

Cos(value) *Returns the math COS() function Example: [i]$x = Cos(27)*

References

ControlCommand Values You can receive values for controls using using the GetControlValue() command. The syntax for this command can be found above in this document. Each control will accept a different collection of commands, which are listed here.

All Controls Pause

Casting Functions These functions are to cast objects from one datatype to another To Double: CDBL() To Integer: CInt() To Long: CLong() To Unsigned Integer: CUint() To Unsigned Long: CULong() To DateTime: CDateTime()

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 D11 D12 D13 D14 D15 D16 D17 D18 D19 D20 D21 D22 D23

Dynamixel Servos (protocol v1) for AX/RX/EX (Port D5) Standard hobby servos use a pulse width modulation for setting the position. Optionally, more advanced users may use the higher priced Dynamixel servos for increased position resolution and strength. These servos are connected together in a chain of wiring (each servo connects to the previous servo). Connect the signal wire of the servo chain to the D5 Port of the EZ-B v4. This feature does not work on the EZ-B v3. Â· AX0 Â· AX1 Â· AX2 Â· AX3 Â· AX4 Â· AX5 Â· AX6 Â· AX7 Â· AX8 Â· AX9 Â· AX10 Â· AX11 Â· AX12 Â· AX13 Â· AX14 Â· AX15 Â· AX16 Â· AX17 Â· AX18 Â· AX19 Â· AX20 Â· AX21 Â· AX22 Â· AX23 Â· AX24 Â· AX25 Â· AX26 Â· AX27 Â· AX28 Â· AX29 Â· AX30 Â· AX31 Â· AX32 Â· AX33 Â· AX34 Â· AX35 Â· AX36 Â· AX37 Â· AX38 Â· AX39 Â· AX40 Â· AX41 Â· AX42 Â· AX43 Â· AX44 Â· AX45 Â· AX46 Â· AX47 Â· AX48 Â· AX49 Â· AX50

Dynamixel Servos (protocol v2) for XL-320 & Pro (Port D5) Standard hobby servos use a pulse width modulation for setting the position. Optionally, more advanced users may use the higher priced Dynamixel servos for increased position resolution and strength. These servos are connected together in a chain of wiring (each servo connects to the previous servo). Connect the signal wire of the servo chain to the D5 Port of the EZ-B v4. This feature does not work on the EZ-B v3. â€¢

AXV0 â€¢ AXV1 â€¢ AXV2 â€¢ AXV3 â€¢ AXV4 â€¢ AXV5 â€¢ AXV6 â€¢ AXV7 â€¢ AXV8 â€¢ AXV9 â€¢ AXV10 â€¢ AXV11 â€¢ AXV12 â€¢ AXV13 â€¢ AXV14 â€¢ AXV15 â€¢ AXV16 â€¢ AXV17 â€¢ AXV18 â€¢ AXV19 â€¢ AXV20 â€¢ AXV21 â€¢ AXV22 â€¢ AXV23 â€¢ AXV24 â€¢ AXV25 â€¢ AXV26 â€¢ AXV27 â€¢ AXV28 â€¢ AXV29 â€¢ AXV30 â€¢ AXV31 â€¢ AXV32 â€¢ AXV33 â€¢ AXV34 â€¢ AXV35 â€¢ AXV36 â€¢ AXV37 â€¢ AXV38 â€¢ AXV39 â€¢ AXV40 â€¢ AXV41 â€¢ AXV42 â€¢ AXV43 â€¢ AXV44 â€¢ AXV45 â€¢ AXV46 â€¢ AXV47 â€¢ AXV48 â€¢ AXV49 â€¢ AXV50

Digital Port Outout Baud Rates The digital ports can output Serial, which differs from the UART. Using any digital port as a Serial Output command can be done by using the SendSerial() command. Using a digital port this way does not include an input buffer. For input buffers, use the UARTx functions. 300 1200 2400 4800 9600 19200 38400 57600 115200

DateTime Functions MinDate() MaxDate() MonthName() AddDays() AddMonths() AddYears() AddHours() AddMinute() AddSeconds() FmtNum() FmtDate()

Else Condition tree must be closed with an ENDIF Example: *If (GetDigital(D0) = 1) Print(â€œYesâ€) Else Print(â€œNoâ€) EndIf*

Exec(EXE/Bat File, [parameters]) Executes a windows application or batch file. The second parameter is a list of optional parameters Example:
Exec(â€œC:\Windows\Notepad.exeâ€) Example:
Exec(â€œC:\Windows\Notepad.exeâ€, â€œC:\MyFile.txtâ€)

FileExists(filename) Returns a 1 or 0 if the specified file exists. Example: *$fileExists = FileExists(â€œc:\temp\mylog.txtâ€)*

FileDelete(filename) Deletes a file on your computer Example: *FileDelete(â€œc:\temp\mylog.txtâ€)*

FileReadAll(filename) Returns the entire contents of the specified file. Example: *$contents = FileReadAll(â€œc:\temp\mylog.txtâ€)*

FileReadChar(filename) Returns the next character in the specified file Example: *$char = FileReadChar(â€œc:\temp\mylog.txtâ€)*

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(â€œc:\temp\mylog.txtâ€)*

FileReadEnd(filename) Returns a 1 or 0 if the file has reached the end. Example: *$fileEnd = FileReadEnd(â€œc:\temp\mylog.txtâ€)*

FileReadLine(filename) Returns the next line of the specified filename. Example: *$line = FileReadLine(â€œc:\temp\mylog.txtâ€)*

FileReadLineRandom(filename) Returns a random line within the specified file Example: *$randomLine = FileReadLineRandom(â€œc:\temp\mylog.txtâ€)*

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(â€œc:\temp\mylog.txtâ€)*

FileWrite(filename, contents) Appends text to the specified file. This does not append a new line. Example: *FileWrite(â€œc:\temp\mylog.txtâ€, â€œMy Variable: â€œ + $x)* Example: *FileWrite(â€œc:\temp\mylog.txtâ€, â€œServo Position: GetServo(d2)â€)*

FileWriteLine(filename, contents) Appends text as a new line to the specified file. Example: *FileWriteLine(â€œc:\temp\mylog.txtâ€, â€œMy Variable: â€œ + $x)* Example: *FileWriteLine(â€œc:\temp\mylog.txtâ€, â€œServo Position: GetServo(d2)â€)*

FillArray(variable, default value) Fill an existing array with the default value. Example: *FillArray($myArray, 88)* Example: *FillArray($myArray, â€œdefault valueâ€)*

Forward([speed], [milliSeconds]) Using a Movement Panel Control, this will start your robot in the Forward direction. Optionally, you can provide the speed and/or number of milliseconds to move. You will require at least one movement panel to be configured within the project. This function will control that movement panel. Speed is a number between 0 and 255 Example: *Forward()* Example: *Forward(200)* Example: *Forward(255, 5000)*

GetADC(Port) Returns the 8 Bit ADC value of the specified port Example: *$x = GetADC(adc0)*

GetADC12(Port) Returns the 12 Bit ADC value of the specified port on the EZ-B v4 Example: *$x = GetADC12(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: *$x = GetArraySize(â€œ$myArrayâ€)*

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: *$value = GetAsByte(â€œHâ€)* Example: *$value = GetAsByte($x)*

GetBit(value, bit) Returns the value of the specified bit. The LSB is bit 0, and MSB is bit 7. Example: *$bit = 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: *$value = GetByte(â€œHâ€)* Example: *$value = 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: *$value = GetByteAt(â€œHello Worldâ€, 2)* Example: *$value = 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 brackets. See the example below. Returns the character at the specified index. If the character at the specified position is outside of readable ASCII, you will want to use GetByteAt() or GetByte() functions instead. The Index is zero based, which means 0 (zero) is the first character. Example: *$char = GetCharAt(â€œHello Worldâ€, 2)* Example: *$char = GetCharAt($x, 2)* Example: *$byte = $x[3]* Example: *$byte = $x[$y]*

GetControlValue(windowName, ControlCommandValues) Gets a value from the window by its name. Look further down in this document for available control command parameters under the ControlCommand Values section. Example: *$x = GetControlValue("ADC Graph", â€œpauseâ€)*

GetCPUTemp() Returns the CPU Temperature of the EZ-B v4 Example: *$x = GetCPUTemp()*

GetDigital(Port) Returns the Digital value of the specified port as a 0 or 1

Example: *$x = GetDigital(d0)*

GetPing(trigger port, echo port) Return the Ping HC-SR04 value of the specified port Example: *$x = GetPing(d0, d1)*

GetPWM (digitalPort) Gets the PWM (Pulse Width Modulation) of specified port PWM is between 0 and 100 Example: *$x = GetPWM(D14)*

GetRandom(Min, Max) Returns a random number between Min and Max Example: *$x = GetRandom(10, 50)*

GetRandomUnique(Min, Max) Returns a random number between Min and Max This function attempts to make the number unique from the last time it was called Example: *$x = GetRandomUnique(10, 50)*

GetServo(Port) Returns the last specified Servo Position value of the servo port. Servo position is between 1 and 180 Example: *$x = GetServo(d0)*

GetServoSpeed(Port) Returns the Servo Speed value of the specified port Speed is between 0 and 10 Example: *$x = GetServoSpeed(d0)*

GetSpeed( ) Returns the global Movement Speed value Speed is between 0 and 255 The speed can be viewed in the Script Variable Viewer Example: *$x = GetSpeed()*

GetSpeedLeft( ) Returns the global Movement Speed value of Left wheel Speed is between 0 and 255 The speed can be viewed in the Script Variable Viewer Example: *$x = GetSpeedLeft()*

GetSpeedRight( ) Returns the global Movement Speed value of Right wheel Speed is between 0 and 255 The speed can be viewed in the Script Variable Viewer Example: *$x = GetSpeedRight()*

GetVoltage() Returns the EZ-B v4 Battery Voltage Example: *$x = GetVoltage()*

GetVolume() Gets the volume of the EZ-B v4 speaker The volume will be a value between 0 (quite), 100 (loud), 200 (2x over drive) The volume can be viewed in the Script Variable Viewer Example: *$x = GetVolume(50)*

GPSStop(latitude, longitude, resolution) Uses the attached GPS control and stops the movement panel when the coordinates are within the specified resolution for the latitude and longitude. Example: *GPSStop(54.01438, -110.4931, 0.0005)*

Goto(label) Goto a specific :Label location Example: *Goto(My_Label)*

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(â€œOneâ€) EndIf* Example: *If ($Day = 2 AND $Hour = 3) Print(â€œHelloâ€) EndIf* Example: *If (GetServo(D5) >20 OR ($x >= 3 and $y , =, !=, AND, OR* Example: *If (GetServo(D0) = 1) Print(â€œOneâ€) ElseIf (GetServo(D0) = 2) Print(â€œTwoâ€) Else Print(â€œSomething Elseâ€) EndIf*

IndexOf(string1, string2) Returns the index within string1 of the first occurrence of string2. This search is case insensitive. Example: *$value = IndexOf(â€œCat In The Hatâ€, â€œInâ€)*

IsConnected(boardIndex) Returns TRUE or FALSE if the specified EZ-B board index is connected Example: *$status = IsConnected(0)*

IsNumeric(value) Returns True or False if the specified value is numeric. Example: *$value = IsNumeric($x)*

I2CClockSpeed(boardIndex, rate) Specify the clock speed of the i2c interface. The default speed is 100000, which is 100khz. Many devices support faster speeds, up to 400000 (400khz). Example: *I2CClockSpeed(0, 100000)* Example: *I2CClockSpeed(0, 400000)*

I2CRead(boardIndex, 7bitDeviceAddress, bytes to expect) Returns a series of ASCII data from the specified address over the i2c interface boardIndex is the EZ-B you wish to use (0 is first EZ-B) Device Hex Address must be hexadecimal 7Bit (0x5e) You must specify the number of Bytes To Expect Example: *$Val = I2CRead(0, 0x5e, 2)*

I2CReadBinary(boardIndex, 7bitDeviceAddress, bytes to expect, variable) Returns a series of binary data from the specified address over the i2c interface into the specified variable array boardIndex is the EZ-B you wish to use (0 is first EZ-B) Device Hex Address must be hexadecimal 7Bit (0x5e) You must specify the number of Bytes To Expect Example: *I2CReadBinary(0, 0x5e, 2, $variable)*

I2CWrite(boardIndex, deviceAddress, data, ...) Send a series of ASCII data to the specified device hex address over the i2c interface. This command will Start i2c, Write Data, and Stop i2c. boardIndex is the EZ-B you wish to use (0 is first EZ-B) Device Hex Address of i2c device must be in 0x00 format. Data can be Hex (0x09), string ("string"), or decimal (188) Example: *I2CWrite(0, 0x09, 0x02, 0x05, 0x06)* Example: *I2CWrite(0, 0x09, 244)* Example: *I2CWrite(0, 0x09, "This is text â€œ + $variable)*

I2CWriteBinary(boardIndex, deviceAddress, variable) Send a series of binary data to the specified device hex address over the i2c interface from the provided variable array. This command will Start i2c, Write Data, and Stop i2c. boardIndex is the EZ-B you wish to use (0 is first EZ-B) Device Hex Address of i2c device must be in 0x00 format. Variable is an array with data that you wish to send Example: *I2CWriteBinary(0, 0x09, $variable)*

Land() Tell your flying drone to land Example: Land()[/i]

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(â€œThis string is 33 characters longâ€)*

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(â€œMyTestâ€)* Example: *LoadProject(â€œMyTest.ezbâ€)* Example: *LoadProject(â€œC:\Temp\MyTest.ezbâ€)*

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, â€œforwardâ€)*

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(â€œFORWARDâ€)*

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) Set the volume of the mp3 trigger between 0 and 255. 0 is louded, 255 is quiet. Example: *MP3TriggerVolume(d0, 38400, 20)*

References

Multiple EZ-B Boards

EZ-Builder supports multiple physical EZ-B Boards connected to your computer. You can specify the board by putting the board number in front of the port. For example: Servo(2.d0, 8) will move the D0 servo on EZ-B board #2 to position 8. If no board index is specified, the first board (zero) is assumed. If using more than one board, the first board is always responsible for movement panels.

Pause( ) Pauses the script at the location where it is called. To resume the script, another script must call the ControlCommandâ€™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(â€œc:\temp\my Audio.mp3â€)*

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(â€œThis is some textâ€)* Example: *Print(â€œToday is $Dayâ€)* Example: *Print(â€œ$pi rounded is Round($pi, 2)â€)*

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(â€œDJ Suresâ€, â€œtestâ€)*

PushVar(NameSpace, Cell, Value) Send the value to the EZ-Cloud. Example: *PushVar(â€œDJ Suresâ€, â€œtestâ€, â€œI am testingâ€)*

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)*

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)*

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(â€œx=â€ + $x) ENDREPEAT*

REPEATUNTIL Repeats the code between REPEATUNTIL and ENDREPEATUNTIL until the specified condition is TRUE. Example: *REPEATUNTIL($second = 30) Print(â€œSecond=â€ + $second) Sleep(500) ENDREPEAT*

REPEATWHILE Repeats the code between REPEATWHILE and ENDREPEATWHILE until the specified condition is FALSE. Example: *REPEATUNTIL($second > 50) Print(â€œSecond=â€ + $second) Sleep(500) ENDREPEAT*

Return() Return from a Goto() If you jump to a position of code with a Goto(), the Return statement will allow you to return back to that piece of code following the last Goto() statement. If you attempt to Return() with an empty stack, nothing will happen. The script will ignore the Return() statement. Example: *Return()*

Reverse([speed], [milliSeconds]) Using a Movement Panel Control, this will start your robot in the Reverse direction. Optionally, you can provide the speed and/or number of milliseconds to move. You will require at least one movement panel to be configured within the project. This function will control that movement panel. Speed is a number between 0 and 255 Example: *Reverse()* Example: *Reverse(200)* Example: *Reverse(255, 5000)*

Right([speed], [milliSeconds]) Using a Movement Panel Control, this will turn your robot right. 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: *Right()* Example #2: *Right(200)* Example #2: *Right(200, 5000)*

RoboQuad(RoboQuadCommand) Send a command to a RoboQuad connected on port D1 on EZ-B 0. Find the available RoboQuad commands further down in this document. Example: *RoboQuad(â€œWalk_Forwardâ€)* Example: *RoboQuad(â€œLeft_Turn_Rollâ€)*

RoboSapien(RoboSapienCommand) Send a command to a RoboSapien connected on port D1 on EZ-B 0. Find the available RoboSapien commands further down in this document. Example: *RoboSapien(â€œWalkForwardâ€)* Example: *RoboSapien(â€œLeftArmUpâ€)*

RollLeft([milliSeconds]) Using the servo port settings from a Movement Panel Control, this will roll your drone robot left Optionally, you can specify the number of milliseconds to turn. Example #1: *RollLeft()* Example #2: *RollLeft(1000)*

RollRight([milliSeconds]) Using the servo port settings from a Movement Panel Control, this will roll your drone robot right Optionally, you can specify the number of milliseconds to turn. Example #1: *RollRight()* Example #2: *RollRight(1000)*

Roomba(cmd) Execute the specified command on a connected Roomba Vacuum on Port D0 and EZ-B 0. Look for the available Roomba commands further below in this document. You may also add the Roomba Movement Panel for graphical controls. Example: *Roomba(â€œinitâ€)* Example: *Roomba(â€œSideBrushOnâ€)*

Round(value, decimals) Returns the math Round() of a number Returns the number rounded to the specified decimal places Example: *$x = Round($pi, 2)* Example: *$x = Round(9.3848291, 1)*

References

RoboQuad Commands Supported on the EZ-B v3 only, the D1 port can be used to connect to the IR wire of some WowWee robots. Due to multiple hardware revisions of the robots, these functions are not always compatible and therefore this feature has been discontinued in the EZ-B v4. Stop Walk_Forward Right_Crab_Walk Left_Crab_Walk Left_Crab_Four_Steps Right_Crab_Four_Steps Backward_Four_Steps Walk_Backward Forward_Four_Steps Rotate_Counter_Clockwise Counter_Clockwise_Four_Steps Rotate_Clockwise Clockwise_Four_Steps Head_Up Head_Down Head_Clockwise Head_Counter_Clockwise Top_Left_Shuffle Top_Right_Shuffle Bottom_Left_Shuffle Bottom_Right_Shuffle Left_Strafe Right_Strafe Left_Turn_Roll Right_Turn_Roll Burst Single_Shot Stomp_Walk Left_Legs_In Left_Legs_Out Left_Forward_Leg_In Left_Forward_Leg_Out Left_Backward_Leg_In Left_Backward_Leg_Out Right_Legs_In Right_Legs_Out Right_Forward_Leg_In Right_Forward_Leg_Out Right_Backward_Leg_In Right_Backward_Leg_Out Program Play_Program Program_Delete_Last_Step Erase_Program Scan_Left_For_Object Scan_Right_For_Object Smart_Scan Approach_Nearest_Object Escape_Walk Toggle_Activity_Level_1 Toggle_Activity_Level_2 Toggle_Activity_Level_3 Toggle_Aggression_1 Toggle_Aggression_2 Toggle_Aggression_3 Toggle_Awareness_1 Toggle_Awareness_2 Toggle_Awareness_3 Leg_Reset Full_Reset Volume_Up Volume_Down Guard Sleep Toggle_Autonomy Toggle_Sensors Twitch Surprise Wave Dizzy Attack Roar Aware_Stance High_Stance Aggressive_Stance Dance_Demo Movement_Demo Leg_Check

RoboSapien Commands Supported on the EZ-B v3 only, the D1 port can be used to connect to the IR wire of some WowWee robots. Due to multiple hardware revisions of the robots, these functions are not always compatible and therefore this feature has been discontinued in the EZ-B v4.

TurnRight RightArmUp RightArmOut TiltBodyRight RightArmDown RightArmIn WalkForward WalkReverse TurnLeft LeftArmUp LeftArmOut TiltBodyLeft LeftArmDown LeftArmIn Stop WakeUp Burp RightHandStrike RightHandSweep RightHandStrike2 HighFive Fart LeftHandStrike LeftHandSweep Whistle Roar

Roomba Commands The iRobot Roomba can be controlled by the EZ-B. Once the Roomba Movement Panel has been added, these commands can be parameters to the Roomba() function. Syntax use for this function can be found above in this document. Init EnableSensors DisableSensors PowerOff SpotClean Clean MaxClean DisableAllBrushes MainBrushOn MainBrushOff SideBrushOn SideBrushOff VacuumOn

VacuumOff SeekDockingStation

Roomba/Sound Music Notes The iRobot Roomba or EZ-B v4 can be told to play audio tones. They may be simple, but it is fun - here is a list of parameters for the RoombaTone() or SoundNote() functions. The syntax for the function can be found above in this document. Here are the acceptable notesâ€¦ C1 Db1 D1 Eb1 E1 F1 Gb1 G1 Ab1 A1 Bb1 B1 C2 Db2 D2 Eb2 E2 F2 Gb2 G2 Ab2 A2 Bb2

The notes below are not supported with iRobot Roomba and only works with EZ-B v4 SoundNote() C3 Db3 D3 Eb3 E3 F3 Gb3 G3 Ab3 A3 Bb3

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, â€œaâ€)* Example: *SendSerial(d0, 9600, "This is text")* Example: *SendSerial(d0, 9600, â€œHello â€œ + $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(â€œ192.168.0.1â€, 21, â€œHello Worldâ€)* Example: *SendUDP(â€œ192.168.0.1â€, 21, 0x20, 0x21, 0x22, 0x30)* Example: *SendUDP(â€œ192.168.0.1â€, 21, 0x20, 0x21, 0x22, â€œHelloâ€)* Example: *SendUDP(â€œ192.168.0.1â€, 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)*

SetSpeedLeft(speed) Sets the global Movement Speed value of the Left wheel Speed is between 0 (slow) and 255 (fast) The speed can be viewed in the Script Variable Viewer Example: *SetSpeedLeft(50)*

SetSpeedRight(speed) Sets the global Movement Speed value of the Right wheel Speed is between 0 (slow) and 255 (fast) The speed can be viewed in the Script Variable Viewer Example: *SetSpeedRight(50)*

StopEZBAudio() Stops any audio background that is being streamed through the EZ-B v4 Example: *StopEZBAudio()*

SayEZBWait(text to speech) Speaks the text that is specified within the brackets out of the EZ-B v4 speaker and blocks until done speaking. Example: *SayEZBWait("Hello, I am a robot")*

Servo (servoPort, position) Move servo to the specified position Servo position is between 1 and 180 Example: *Servo(D14, 25)*

ServoDown (servoPort, count) Decrement the servo position value by specified count Servo position is between 1 and 180 Example: *ServoDown(D14, 1)*

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)*

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)*

SetVolume(volume) Sets the volume of the EZ-B v4 speaker The volume can be a value between 0 (quite), 100 (loud), 200 (2x over drive) The volume can be viewed in the Script Variable Viewer Example: *SetVolume(50)* Example: *SetVolume(100)* Example: *SetVolume(150)*

ServoRandom (servoPort, lowPosition, highPosition) Move the servo to a random position between low and high Servo position is between 1 and 180 Example: *ServoRandom(D14, 10, 20)*

ServoSpeed (servoPort, speed) Set the speed of servo or PWM. This is the speed to move between positions. The servo speed is a number between 0 (fastest) and 10 (slowest) ***Note:** To initialize the ServoSpeed() at first use, set a Servo() position before using the ServoSpeed() command. If there is no previous position (such as during power-on), the software assumes the position is 0 and will cause issues with your robot. ***Note:** Once the ServoSpeed() has been initialized the first time, specify the ServoSpeed() before specifying the Servo() position. Example: *ServoSpeed(D14, 25)*

ServoSpeedRandom (servoPort, lowSpeed, highSpeed) Set the servo speed or PWM to a random value The servo speed is a number between 0 (fastest) and 10 (slowest) ***Note:** To initialize the ServoSpeed() at first use, set a Servo() position before using the ServoSpeed() command. If there is no previous position (such as during power-on), the software assumes the position is 0 and will cause issues with your robot. ***Note:** Once the ServoSpeed() has been initialized the first time, specify the ServoSpeed() before specifying the Servo() position. Example: *ServoSpeedRandom(D14, 10, 20)*

ServoUp (servoPort, count) Increment the servo position value by specified count Servo position is between 1 and 180 Example: *ServoUp(D14, 1)*

Servo_Wait (digitalPort, higher/lower/equals, value) Wait until the Servo Port is higher or lower than specified value. Zero can be specified as a value for a stopped servo. Example: *Servo_Wait(D5, HIGHER, 20)*

Set (digitalPort, on/off/true/false) Set a digital port state to either on or off Example: *Set(D2, OFF)*

SetRandom (digitalPort) Set a digital port to a random state of either on or off

Example: *SetRandom(D2)*

ShowControl(ControlName) Used for mobile devices and the Interface Builder only, this command will open the specified control into the foreground. Example: *ShowControl(â€œWii Remoteâ€)*

ShowDesktop(desktopNumber) Shows the specified virtual desktop. The desktop number is 1, 2 or 3. Example: *ShowDesktop(1)*

Sin(value) Returns the math SIN() function Example: *$x = Sin(27)*

Sleep (milliseconds) Pauses for specified milliseconds Example sleeps for 1 second: *Sleep(1000)*

SleepPC(Suspend|Hibernate, force, wake) Sends a command to the operating system to sleep or hibernate. If Force is TRUE, the computer is forced to sleep and other applications have no say in the decision. If Wake is TRUE, the computer will wake up on Wake events. Example: *SleepPC(Suspend, true, true)* Example: *SleepPC(Hibernate, true, true)*

SleepRandom (lowMilliSec, highMilliSec) Pauses for a random millisecond delay between the 2 provided values Example: *SleepRandom(1000, 5000)*

SoundNote(note, lengthMS, [signal type]) Plays the specified audio note out of the EZ-B v4 speaker for the specified number of milliseconds. Optionally, you may provide a signal type as well. The valid signal types are Sine, Square, Triangle, Pulse, Sawtooth, WhiteNoise, GaussNoise, DigitalNoise. Example: *SoundNote(â€œC2â€, 1000)* Example: *SoundNote(â€œC2â€, 1000, â€œSquareâ€)*

SpeakRSS(url, [story index]) Speaks the title and text of the rss url out of the PC Sound Card. Example #1: *SpeakRSS("http://rss.cbc.ca/lineup/world.xml")* Example #2: *SpeakRSS("http://rss.cbc.ca/lineup/world.xml", 3)*

SpeakRSSDescription(url, [story index]) Speaks only the text of the rss url out of the PC Sound Card. Example #1: *SpeakRSSDescription("http://rss.cbc.ca/lineup/world.xml")* Example #2: *SpeakRSSDescription("http://rss.cbc.ca/lineup/world.xml", 3)*

SpeakStop( ) Stops speaking the current specified phrases out of the PC Sound Card. Example: *SpeakStop()*

SpeakTwitter(twitterUserName, [story index]) Speaks the twitter feed for the specific username out of the PC Sound Card. Example #1: *SpeakTwitter("EZ_Robot")* Example #2: *SpeakTwitter("EZ_Robot", 3)*

SpeakVolume(value) The volume of the speech synthesizer out of the PC Sound Card. The value is between 0 and 100 Example: *SpeakVolume(30)*

Split(text, splitChar, index) Splits a line of text by the specified SplitChar into an array and returns the Index. The Index is zero based, which means 0 (zero) is the first item within the array. Example: *$contents = Split(â€œOne, Two, Threeâ€, â€œ,â€, 1)* Example: *$contents = Split(â€œOne - Two - Threeâ€, â€œ-â€, 2)*

Stop() Using Movement Panel Control, this will stop your robot. You will require at least one movement panel to be configured within the project. This function will

control a movement panel. Example: *Stop()*

StopAudio() Stops the current audio file that is playing through the default audio device by PlayAudio() Example: *StopAudio()*

Sqrt(value) Returns the math Square Root function Example: *$x = Sqrt(9)*

SubString(string1, start, length) Returns the specified substring within string1. Example: *$value = SubString(â€œCat In The Hatâ€, 4, 2)*

References

Servo/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 D11 D12 D13 D14 D15 D16 D17 D18 D19 D20 D21 D22 D23

Scientific Math Functions Sin() Cos() Tan() Sec() Csc() Cot() ASin() ACos() ATan() SinH() CosH() TanH() Abs() Sqrt() Ciel() Floor() Exp() Log10() Log() Max() Min() Round() E() Pi() Now() Today()

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 â€œ0x39â€. 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 Pinâ€™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 Pinâ€™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 Pinâ€™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 Pinâ€™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 Pinâ€™s associated with each UART Port. Example: *UARTReadBinary(0, 0, 10, $variable)* Example: *UARTReadBinary(0, 0, UARTAvailable(0, 1), $variable)*

UARTWrite(boardIndex, port, data) Write ASCII data through the Peripheral UART. 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 Pinâ€™s associated with each UART Port. Example: *UARTWrite(0, 0, â€œhello worldâ€)* Example: *UARTWrite(0, 0, 0x30, 0x40, â€œhelloâ€)*

UARTWriteBinary(boardIndex, port, variable) Write binary variable array data through the Peripheral UART. 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 Pinâ€™s associated with each UART Port. Example: *UARTWriteBinary(0, 0,*

$variable)

References

UART Ports The UARTx ports are used connect to Serial TTL devices for both input and output. Contrary to the digital port Serial Output, these peripherals will also receive data into an input buffer as well. The input buffer of each UART is 5,000 Bytes. There are 3 UARTs, the first is the hardware labelled port, second and third are digital pins. These UARTs are controlled using the UARTInit(), UARTWrite(), UARTRead() and UARTAvailable() commands. The speed of these UARTs can be any integer between 1 and 3750000 bps. Â· UART0 TX: Expansion Connector Â· UART0 RX: Expansion Connector Â· UART1 TX: D5 Â· UART1 RX: D6 Â· UART2 TX: D18 Â· UART2 RX: D19

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 = â€œThis is a stringâ€ String Concat: $str = â€œfuâ€ + $bar Integer: $number = 6 Floating Point: $dec = 3.14 Byte: $byte = 0x52 Boolan: $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 $month $year $day $dayName $hour $minute $second $monthName $time $pi

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 â€œtimeoutâ€ if no word is detected in the specified timeout length. Example: *WaitForSpeech(30, â€œYesâ€, â€œNoâ€)* Example: *$value = WaitForSpeech(30, â€œYesâ€, â€œNoâ€)*

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)*

b# Commented Text[/b] Comment a line of code Example: *# This is a comment. This code will not run*

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 and will be made when possible.

Happy building (and scripting).

Manual contents correct at time of writing, (16th October 2015).