

eigerScript


Programming Language for eigerPanels



Creation: 14.01.2011 / CA & SL
Update: 31.10.2011 / CA

eigerGraphics



FOX embedded computers 
the canny swiss solution

developed and produced by



S-TEC electronics AG

Industriestrasse 49
CH-6300 Zug
Switzerland
www.s-tec.ch

www.eigergraphics.com

Contents:

List of eigerScript Methods	2
Built-In Functions	2
Class: Binary	3
Class: Buffer	3
Class: Buzzer	3
Class: Colors_15	4
Class: CSV	5
Class: DataServer	5
Class: Debug	6
Class: Delay	6
Class: Display	6
Class: Draw	7
Class: EEPROM	8
Class: EVE	8
Class: File	8
Class: Fill	8
Class: HotKey	8
Class: HotSpot	9
Class: I2C	9
Class: InOut	10
Class: Label	10
Class: Load	10
Class: Math	11
Class: Serial	12
Class: Sound	12
Class: Store	12
Class: Str (String)	12
Class: System	14
Class: Time	14
Class: Timer	14
Class: Transfer	14
Class: Value	15
Support	15

■ List of eigerScript Methods

Legend:

X Methods described in the [eigerPanel-SoftwareManual](#)

Built-In Functions

	CallBack(VarLong:CallBackAddress)	Call subroutine stored in var CallBackAddress
	CallSubroutine(labelRelative24:SubroutineName)	Call a subroutine SubroutineName auf
	Experimental.ADD_STACK_Integer()	
	Experimental.NOP()	
	GotoProject(VarStr:ProjectName,VarInt:ViewID)	wechselt ins Projekt

		WXYZ und ViewID 123
	GotoView(VarInt:ViewID)	wechselt zur View mit der Nummer ViewID
	Jump(labelRelative24)	Jump to a label
	Jump_IF_BitClear_Integer(labelRelative24:SprungDest,VarInt:TestInteger,VarInt:BitNummer)	
	Jump_IF_BitSet_Integer(labelRelative24:SprungDest,VarInt:TestInteger,VarInt:BitNummer)	
	Jump_IF_false(labelRelative24)	
	Jump_IF_Stack_false(labelRelative24)	
	Jump_IF_Stack_true(labelRelative24)	
	Jump_IF_true(labelRelative24)	
	TestInteger(VarInt)	

Class: Binary

	eigerScript Methods	Short Description
x	Binary.AND_Integer(VarInt:Q,VarInt:A,VarInt:B)	Q = A AND B (binary)
x	Binary.AND_Long(VarLong:Q,VarLong:A,VarLong:B)	Q = A AND B (binary)
x	Binary.BCLR_Integer(VarInt:Dest,VarInt:BitNo)	Bit Clear (BitNo = 0..15)
x	Binary.BCLR_Long(VarLong:Dest,VarInt:BitNo)	Bit Clear (BitNo = 0..31)
x	Binary.BGET_Integer(VarInt:SrcVar,VarInt:BitNo,VarInt:Result)	Get Bit (BitNo = 0..15)
x	Binary.BGET_Long(VarLong:SrcVar,VarInt:BitNo,VarInt:Result)	Get Bit (BitNo = 0..31)
x	Binary.BNOT_Integer(VarInt:Dest,VarInt:BitNo)	Negate Bit (BitNo = 0..15)
x	Binary.BNOT_Long(VarLong:Dest,VarInt:BitNo)	Negate Bit (BitNo = 0..31)
x	Binary.BSET_Integer(VarInt:Dest,VarInt:BitNo)	Set Bit (BitNo = 0..15)
x	Binary.BSET_Long(VarLong:Dest,VarInt:BitNo)	Set Bit (BitNo = 0..31)
x	Binary.NOT_Integer(VarInt:Q,VarInt:A)	Q = NOT A (binary)
x	Binary.NOT_Long(VarLong:Q,VarLong:A)	Q = NOT A (binary)
x	Binary.OR_Integer(VarInt:Q,VarInt:A,VarInt:B)	Q = A OR B (binary)
x	Binary.OR_Long(VarLong:Q,VarLong:A,VarLong:B)	Q = A OR B (binary)
	Binary.ROTATE_LEFT_Integer(VarInt:Q,VarInt:NumberOfBits)	Rotate Bits left
	Binary.ROTATE_RIGHT_Integer(VarInt:Q,VarInt:NumberOfBits)	Rotate Bits right
x	Binary.SWAP_BYTE_Integer(VarInt:Q,VarInt:A)	Swap lower and higher bytes of A and place result in Q
x	Binary.SWAP_WORD_Long(VarLong:Q,VarLong:A)	Swap lower and higher Words of A and place result in Q
x	Binary.XOR_Integer(VarInt:Q,VarInt:A,VarInt:B)	Q = A XOR B (binary)
x	Binary.XOR_Long(VarLong:Q,VarLong:A,VarLong:B)	Q = A XOR B (binary)

Class: Buffer

	Buffer.Get_INT16(VarInt:INT16,VarStr:Buffer,VarLong:Offset)	
	Buffer.Get_INT32(VarLong:INT32,VarStr:Buffer,VarLong:Offset)	
	Buffer.Get_INT8_I(VarInt:INT8,VarStr:Buffer,VarLong:Offset)	
	Buffer.Get_UINT16(VarInt:UINT16,VarStr:Buffer,VarLong:Offset)	
	Buffer.Get_UINT8_I(VarInt:UINT8,VarStr:Buffer,VarLong:Offset)	
	Buffer.Put_UINT8_I(VarInt:UINT8,VarStr:Buffer,VarLong:Offset)	

Class: Buzzer

	Buzzer.Alarm(VarInt:ON_Time,VarInt:OFF_Time,VarInt:Pulse,VarInt:Pause,VarInt:Loops)	Buzzer Alarmpattern
	Buzzer.Click()	Buzzer Click
	Buzzer.Off()	Buzzer OFF

	Buzzer.On()	Buzzer ON
	Buzzer.Timer(VarInt:Time)	Buzzer for Time ms on
	Buzzer.Touch(VarInt:Time)	Buzzer on at every Touchdown

Class: Colors_15

x	Colors_15.AutoColor(VarInt:DestCol15,VarInt:SrcCol15)	returns black or white with maximal contrast
x	Colors_15.BlackWhite(VarInt:DestCol15,VarInt:SrcCol15)	returns black for dark and white for light colors
x	Colors_15.Brighten(VarInt:DestCol15,VarInt:SrcCol15)	returns the brightened color of the original color
x	Colors_15.ColorMix(VarInt: DestCol15, VarInt: Src_1,VarInt:Src_2)	returns the 1 to 1 color mix of the input colors
x	Colors_15.ColorMix_3to1(VarInt:DestCol15,VarInt:Src_3P,VarInt:Srcr_1P)	returns the 3:1 color mix of the input colors
x	Colors_15.Darken(VarInt:DestCol15,VarInt:SrcCol15)	returns the darkened color of the original color
x	Colors_15.GetBlue(VarInt:Blue_Component,VarInt:SrcCol15)	returns the blue component of a color in the range of 0..31
x	Colors_15.GetGreen(VarInt:Green_Component,VarInt:SrcCol15)	returns the green component of a color in the range of 0..31
x	Colors_15.GetPixelColor(VarInt:DestCol15,VarInt:X,VarInt:Y)	gets the color of the pixel at coordinate X,Y
x	Colors_15.GetRed(VarInt:Red_Component,VarInt:SrcCol15)	returns the red component of a color in the range of 0..31
x	Colors_15.GreyValue(VarInt:DestCol15,VarInt:SrcCol15)	returns the greyvalue of a original color
	Colors_15.InverseColor(VarInt:DestCol15,VarInt:SrcCol15)	returns the complementary color of the source
x	Colors_15.LessBlue(VarInt:Color15)	decrements the blue channel
x	Colors_15.LessBlueLimit(VarInt:Color15)	decrements the blue channel until min
x	Colors_15.LessGreen(VarInt:Color15)	decrements the green channel
x	Colors_15.LessGreenLimit(VarInt:Color15)	decrements the green channel until min
x	Colors_15.LessRed(VarInt:Color15)	decrements the red channel
x	Colors_15.LessRedLimit(VarInt:Color15)	decrements the red channel until min
x	Colors_15.Load_RGB(VarInt: Col15,VarInt:Red,VarInt:Green,VarInt:Blue)	converts the tree color components R G B to a COLOR15 variable
x	Colors_15.MoreBlue(VarInt:Color15)	increments the blue channel
x	Colors_15.MoreBlueLimit(VarInt:Color15)	increments the red channel until max
x	Colors_15.MoreGreen(VarInt:Color15)	increments the green channel
x	Colors_15.MoreGreenLimit(VarInt:Color15)	increments the red channel until max
x	Colors_15.MoreRed(VarInt:Color15)	increments the red channel

x	Colors_15.MoreRedLimit(VarInt:Color15)	increments the red channel until max
x	Colors_15.SetColorPalette(VarInt:SrcColor5B)	sets the colorpalette
	Colors_15.SetGreyPalette(VarInt:SrcColor5B)	sets the greypalette
	Colors_15.SetMixPalettes(VarInt:SrcColor5B)	sets the mixpalettes
x	Colors_15.Swap_Blue_Red(VarInt:DestColor5B,VarInt:SrcColor5B)	swaps the blue and the red color component
x	Colors_15.Swap_Green_Blue(VarInt:DestColor5B,VarInt:Src_Color5B)	swaps the green and the blue color component
x	Colors_15.Swap_Red_Green(VarInt:DestColor5B,VarInt:Src_Color5B)	swaps the red and the green color component

Class: CSV

Note: Before using following analysis methods you need to read your CSV file into a string: File.Read_CSV(VarStr:FileName,VarStr:Buffer) . The analysis methods must refer to that string.

	CSV.Analyze_String(VarStr:CSV)	Analyze a string prior to be used with the different CSV methods.
	CSV.DataFieldLength(VarInt:Length,VarStr:CSV, VarInt:Row, VarInt:Col)	Get the length of a field (number of chars)
x	CSV.Find_in_Column(VarStr:CSV,VarInt:StartRow,VarInt:Col, VarStr:SearchStr,VarInt:LineNoFound,VarStr:DataRow)	Find the string 'SearchStr' in the column 'Col' of the CSV-String 'CSV' starting from row 'StartRow' and set 'LineNoFound' with the row number found and 'DataRow' with a copy of the entire row.
	CSV.Get_ByteHex(VarInt:Number,VarStr:CSV,VarInt:Row,VarInt:Col)	Get a byte hex value (e.g. 3B)
	CSV.Get_HighColor(VarInt:Color,VarStr:CSV,VarInt:Row,VarInt:Col)	Get a color (e.g. #25AAFF)
	CSV.Get_Integer(VarInt:Number,VarStr:CSV,VarInt:Row,VarInt:Col)	Get an integer value
	CSV.Get_Long(VarLong:Number,VarStr:CSV,VarInt:Row,VarInt:Col)	Get a long value
	CSV.Get_LongDeci(VarLong:Number,VarStr:CSV,VarInt:Row,VarInt:Col,VarInt:NoDecimals)	Get a long deci value with a given number of decimals
	CSV.Get_LongHex(VarLong:Number,VarStr:CSV,VarInt:Row,VarInt:Col)	Get a long hex value (e.g. 2F04AA3B)
	CSV.Get_String(VarStr:String,VarStr:CSV,VarInt:Row,VarInt:Col)	Get a string
	CSV.Get_UInteger(VarInt:Number,VarStr:CSV,VarInt:Row,VarInt:Col)	Get an unsigned integer (not implemented in eigerStudio 1.03)
	CSV.Get_WordHex(VarInt:Number,VarStr:CSV,VarInt:Row,VarInt:Col)	Get a word hex value (e.g. 2F03)
x	CSV.GetMax_Columns(VarInt:MaxColumns,VarStr:CSV)	Get the max. number of columns
x	CSV.GetMax_Lines(VarInt:MaxLines,VarStr:CSV)	Get the max. number of rows
	CSV.Put_String(VarStr:String,VarStr:CSV,VarInt:Row,VarInt:Col)	Store a string

Class: DataServer

	DataServer.ReceiveToFile(VarInt:NodeAddress,VarStr:FileName)	Receive a file from a peripheral node
	DataServer.ReceiveToString(VarInt:NodeAddress,VarStr:DestString)	Receive a string from a peripheral node

DataServer.Rx_ReadInteger(VarInt:NodeAddress,VarInt:Register,VarInt:Value)	reads an integer value from a peripheral node
DataServer.Rx_ReadLong(VarInt:NodeAddress,VarInt:Register,VarLong:Value)	reads a long value from a peripheral node
DataServer.Rx_String(VarInt:NodeAddress,VarInt:RegNo,VarStr:String)	reads a string from a peripheral node
DataServer.SendFromFile(VarInt:NodeAddress,VarStr:FileName)	Send a file to a peripheral node
DataServer.SendFromString(VarInt:NodeAddress,VarStr:SourceString)	Send a string to a peripheral node
DataServer.Tx_Command(VarInt:NodeAddress,VarInt:Command)	writes a command to a peripheral node
DataServer.Tx_String(VarInt:NodeAddress,VarInt:RegNo,VarStr:String)	writes a string to a peripheral node
DataServer.Tx_WriteInteger(VarInt:NodeAddress,VarInt:Register,VarInt:Value)	writes an integer value to a peripheral node
DataServer.Tx_WriteLong(VarInt:NodeAddress,VarInt:Register,VarLong:Value)	writes a long value to a peripheral node

Class: Debug

Debug.DumpBytes(VarInt)	
Debug.Mark(str)	
Debug.Print_Char(VarInt)	
Debug.Print_CRLF()	
Debug.Print_IntegerHex(str,VarInt)	
Debug.Print_LongHex(str,VarLong)	
Debug.Print_SingleHex(str,VarSingle)	
Debug.Print_String(VarStr)	
Debug.Print_StringHeader(VarStr)	

Class: Delay

Delay.Wait_n_ms(VarInt:Time_ms)	Delays the execution of the program
---------------------------------	-------------------------------------

Class: Display

x	Display.BacklightIntensity_Set(VarInt:Intensity)	Sets the intensity of the backlight 0..1000
x	Display.Clear()	fills Display with Color in el.DisplayColor
x	Display.ClearColor(VarInt)	fills Display with given Color
	Display.CopyLineDown()	
	Display.CopyLineDownLength(VarInt:Length)	
	Display.CopyLineRight()	
	Display.CopyLineRightLength(VarInt:Length)	
x	Display.CopyWindow()	copies a window of the display (X1,X2,W,H) to another position (X2,Y2) of the display
x	Display.Direct()	Display updates change the visible RVR VideoBuffer
	Display.Normal()	Display updates change the hidden AVR and the visible RVR VideoBuffer

x	Display.Prepare()	Display updates change the hidden AVR VideoBuffer
	Display.RestoreWriteMode()	
x	Display.Show()	Copys the hidden AVR VideoBuffer to the visible RVR VideoBuffer
x	Display.ShowLineDown()	
	Display.ShowLineDownLength(VarInt:Length)	
	Display.ShowLineRight()	
	Display.ShowLineRightLength(VarInt:Length)	
	Display.ShowTranslateWindow()	
	Display.ShowWindow()	Copies the Window with geometry given in el.-Registers vom AVR to RVR VideoBuffer
	Display.ShowWindowEffect(VarInt:ShowWindowEffect,VarInt:Speed)	Copies AVR to RVR with given Effect and given Speed
	Display.WriteCommand(VarInt)	

Class: Draw

	Draw.Border()	
	Draw.BorderNumber(VarInt:BorderStyle)	
	Draw.Char(VarInt:Character)	
x	Draw.Circle()	
	Draw.CircleFilled()	
	Draw.ColorVector_Down(labelRelative24:ColorVector)	
	Draw.ColorVector_Right(labelRelative24:ColorVector)	
	Draw.ColorVector_Up(labelRelative24:ColorVector)	
x	Draw.Ellipse()	
	Draw.EllipseFilled()	
	Draw.Icon()	
	Draw.IconNumber(VarInt:Icon_)	
	Draw.Line()	
	Draw.LineDown()	
	Draw.LineDownLength(VarInt:Length)	
	Draw.LineLeft()	
	Draw.LineLeftLength(VarInt:Length)	
	Draw.LineRight()	
	Draw.LineRightLength(VarInt:Length)	
	Draw.LineUp()	
	Draw.LineUpLength(VarInt:Length)	
	Draw.Mark1Round()	
	Draw.Mark1Square()	
	Draw.Mark2Round()	
	Draw.Mark2Square()	
x	Draw.Pixel()	
	Draw.PixelColor(VarInt:Color)	
x	Draw.Rectangle()	
x	Draw.RectangleFilled()	
	Draw.RectangleFilledRound(VarInt:RoundCorner_)	
	Draw.SeparatorLine()	
	Draw.Symbol()	
	Draw.Symbol_DeltaUL(VarInt:Symbol,VarInt:Delta_X,VarInt:Delta_Y)	
	Draw.SymbolNumber(VarInt:Symbol_)	

Class: EEPROM

	EEPROM.Read_Integer(VarInt:EEPROM_Address,VarInt:Variable)	
	EEPROM.Read_Long(VarInt:EEPROM_Address,VarLong:Variable)	
	EEPROM.Read_Single(VarInt:EEPROM_Address,VarSingle:Variable)	
	EEPROM.Read_String(VarInt:EEPROM_Address,VarStr:DestString)	
	EEPROM.Write_Integer(VarInt:EEPROM_Address,VarInt:Variable)	
	EEPROM.Write_Long(VarInt:EEPROM_Address,VarLong:Variable)	
	EEPROM.Write_Single(VarInt:EEPROM_Address,VarSingle:Variable)	
	EEPROM.Write_String(VarInt:EEPROM_Address,VarStr:SourceString,VarInt:NoOfChar)	

Class: EVE

	EVE.Init()	
x	EVE.Load_Transparence()	
x	EVE.Load_XSTART(VarInt:X-Pixels)	Sets the horizontal start of the viewable area of the video buffer
x	EVE.Process_Transparence()	
x	EVE.WriteEnable_AVR()	
x	EVE.WriteEnable_AVR_RVR()	
x	EVE.WriteEnable_RVR()	

Class: File

	File.AppendString(VarStr:FileName,VarStr:String)	
	File.Close(VarInt:Handle.I)	
x	File.DeleteFile(VarStr:FileName)	
	File.FileExists(VarStr:FileName,VarInt:Boolean)	
	File.FileSize(VarStr:FileName,VarLong:FileSize)	
	File.OpenAppendWrite(VarInt:Handle.I,VarStr:FileName)	
	File.OpenRead(VarInt:Handle.I,VarStr:FileName)	
	File.OpenWrite(VarInt:Handle.I,VarStr:FileName)	
	File.Read_CSV(VarStr:FileName,VarStr:Buffer)	
x	File.Read_EGI(VarStr:FileName)	
	File.Read_TextFile(VarStr:FileName,VarStr:Buffer)	
	File.Read_ToBuffer(VarInt:Handle.I,VarStr:Buffer,VarLong:Offset,VarLong:NumBytes,VarLong:BytesRead)	
	File.ScreenDump(VarStr:FileName)	
	File.Write_TextFile(VarStr:FileName,VarStr:Buffer)	

Class: Fill

x	Fill.LabelParameter(labelRelative24)	loads a label structure. Order in structure is very important!
---	--------------------------------------	--

Class: HotKey

x	HotKey.DeInstallKey(VarInt:Key)	DeInstalls a KeyEvent in the KeyEventTable
x	HotKey.DisableLocalKeys()	disables local KeyEvents
x	HotKey.EnableLocalKeys()	enables local KeyEvents
x	HotKey.InputFlush()	Clears the input Buffer of the Key-Inputs
x	HotKey.InstallLocalKey(VarInt:Key,labelRelative24:Event,VarInt:Tag)	Installs a local KeyEvent
x	HotKey.TableDisable()	Disables the

		KeyEventTable
x	HotKey.TableEnable()	Enables the KeyEventTable
x	HotKey.TableInit()	Clears all Events in the KeyEventTable

Class: HotSpot

	HotSpot.DeInstall()	DeInstalls a HotSpot from HotSpotEventTable
x	HotSpot.DeInstallGroup()	DeInstalls a HotSpots with GroupNumber from HotSpotTable
	HotSpot.Disable_BaseGroup()	Disables HotSpots with GroupNumber 1..15
x	HotSpot.Disable_By_ID(VarInt:HotSpot_ID)	Disables HotSpot with given ID
x	HotSpot.DisableGroup(VarInt:Group)	Disables HotSpots with given GroupNumber
x	HotSpot.DisableRegion()	Disables HotSpots in the rectangular Region given in the el-Registers
	HotSpot.Enable_BaseGroup()	Enables HotSpots with GroupNumber 1..15
x	HotSpot.Enable_By_ID(VarInt:HotSpot_ID)	Enables HotSpot with given ID
x	HotSpot.EnableGroup(VarInt:Group)	Enables HotSpots with given GroupNumber
x	HotSpot.EnableRegion()	Enables HotSpots in the rectangular Region given in the el-Registers
x	HotSpot.GetCurrGroupNumber()	Gets the current GroupNumber
x	HotSpot.GetInfo_By_ID(VarInt:HotSpot_ID)	Reads Content of the HotSpot given by ID to the el.HS_-Registers
x	HotSpot.GetNextGroupNumber()	Gets the next free GroupNumber
x	HotSpot.Install(labelRelative24:Enter, labelRelative24:Leave, labelRelative24:Down, labelRelative24:Up)	
x	HotSpot.TableDestroy()	Destroys the whole HotSpotEventTable
	HotSpot.TableDisable()	Disables HotSpot Events
x	HotSpot.TableEnable()	Enables HotSpot Events

Class: I2C

x	I2C.Init()	□initialises the I2C-Master
x	I2C.ReadLastByte(VarInt:Value)	I2C-Master reads last or single Byte
x	I2C.ReadStreamByte(VarInt:Value)	I2C-Master reads a Byte in a Stream
x	I2C.RepeatedStart()	I2C-Master sends RepeatetStart Sr sequence
x	I2C.SetMode(VarInt:I2C_MODE_)	Sets the I2C Datarate
x	I2C.Start()	I2C-Master sends Start S sequence

x	I2C.Stop()	I2C-Master sends Stop P sequence
x	I2C.WriteByte(VarInt:Value)	I2C-Master sends a Byte
x	I2C.WriteConfigByte(VarInt:SlaveAddress,VarInt:I2C_DATADIRECTION_)	I2C-Master sends SlaveAddress and the data direction

Class: InOut

	InOut.DA_Out_8(VarInt:Output_OP93_OP94,VarInt:Value_0-255)	
	InOut.DigitalOutputDriver(VarInt:Kanal Output_OP90,VarInt:Output_Function)	
	InOut.DownCounter_Get(VarInt:Input_IP91,VarLong:Counter)	
	InOut.DownCounter_Load(VarInt:Input_IP91,VarLong:Counter)	
	InOut.DownCounter_Zero_InstallLocal(labelRelative24:Event)	
x	InOut.PWM_Out(VarInt:Output_OP72_OP76,VarInt:Value_0-1000)	
x	InOut.Read_ADC(VarInt:Kanal,VarInt:ADC-Value_0-1023)	
x	InOut.Read_Input(VarInt:Input_IP91,VarInt:Value)	
	InOut.UpCounter_Get(VarInt:Input_IP91,VarLong:Counter)	
	InOut.UpCounter_Reset(VarInt:Input_IP91)	

Class: Label

x	Label.Box()	renders a label with without text
x	Label.Color(VarInt:COLOR)	set the label colors
	Label.Font_Init()	Initialises the font set in el.FontNumber
x	Label.GreyColor()	sets the label colors to grey values
x	Label.PrintString(VarStr:String)	renders a string on a label
	Label.SetFont(VarStr)	set embedded user font
x	Label.StringWidth(VarInt:Width,VarStr:String)	computes the width in pixel of a string
x	Label.Text(VarStr:Text)	renders a label with text

Class: Load

	Load.CallBackAddress(VarLong:CallBackAddress,labelRelative24:SubroutineAddress)	installs a callback routine
x	Load.Color_BT(VarInt:BackColor,VarInt:TextColor)	loads the el.BackColor and the el.TextColor registers
x	Load.Color_FL(VarInt:FillColor,VarInt:LineColor)	loads the el.FillColor and the el.LineColor registers
	Load.ColorScheme(VarInt:ColorScheme_Number)	
	Load.Data_from_ARRAY_Integer(labelRelative24,VarInt:Index,VarInt:Value)	
	Load.Data_from_ARRAY_Long(labelRelative24,VarInt:Index,VarLong:Value)	
	Load.Data_from_ARRAY_Sbyte(labelRelative24,VarInt:Index,VarInt:Value)	
	Load.Data_from_ARRAY_Single(labelRelative24,VarInt:Index,VarSingle:Value)	
	Load.Data_from_ARRAY_Ubyte(labelRelative24,VarInt:Index,VarInt:Value)	
	Load.Enlarge(VarInt:Number)	number in pixels
x	Load.Geometry_XYWH(VarInt:X1,VarInt:Y1,VarInt:W,VarInt:H)	loads the el.Pos_X1, the el.Pos_Y1, the el.Width and the el.Height registers

x	Load.Offset_XY(VarInt:X,VarInt:Y)	loads the el.Offset_X and the el.Offset_Y registers
x	Load.Pos_X1Y1(VarInt:X1,VarInt:Y1)	loads the el.Pos_X1 and the el.Pos_Y1 registers
x	Load.Pos_X2Y2(VarInt:X2,VarInt:Y2)	loads the el.Pos_X2 and the el.Pos_Y2 registers
x	Load.Width_Height(VarInt:W,VarInt:H)	loads the el.Width and the el.Height registers

Class: Math

	Math.ABS_Integer(VarInt:Y,VarInt:A)	
	Math.ABS_Long (VarLong:Y,VarLong:A)	
	Math.ADD_Integer(VarInt,VarInt,VarInt)	
	Math.ADD_IntegerTest(VarInt,VarInt,l,b)	
	Math.ADD_Long(VarLong,VarLong,VarLong)	
	Math.ADD_Single(VarSingle,VarSingle,VarSingle)	
	Math.COS_Single(VarSingle:Y,VarSingle:Winkel_RAD)	
x	Math.CVT_Integer_from_Long(VarInt:Integer,VarLong:Long)	
x	Math.CVT_Long_from_Integer(VarLong:Long,VarInt:Integer)	
x	Math.CVT_Long_from_Single(VarLong:Long,VarSingle:Single)	
x	Math.CVT_Long_from_Single_Round(VarLong,VarSingle)	
x	Math.CVT_Long_from_Uinteger(VarLong:Long,VarInt:Integer)	
x	Math.CVT_LongDeci_from_Single(VarLong:Long,VarSingle:Single,VarInt:D ezimalstellen)	
x	Math.CVT_Single_from_Integer(VarSingle:Single,VarInt:Integer)	
x	Math.CVT_Single_from_Long(VarSingle:Single,VarLong:Long)	
x	Math.CVT_Single_from_LongDeci(VarSingle:Single,VarLong:Long,VarInt:D ezimalstellen)	
	Math.CVT_Single_from_Uinteger(VarSingle:Single,VarInt:Integer)	
	Math.CVT_Uinteger_from_Long(VarInt:Integer,VarLong:Long)	
	Math.DEC_Integer(VarInt:Y)	
	Math.DEC_Long(VarLong)	
	Math.DIV_Integer(VarInt:Y,VarInt:A,VarInt:B)	
	Math.DIV_Long(VarLong,VarLong,VarLong)	
	Math.DIV_Single(VarSingle,VarSingle,VarSingle)	
	Math.EVAL()	
	Math.EXP_Single(VarSingle:Y,VarSingle:X)	
	Math.Formula_Evaluator()	
x	Math.HWRD_from_Long(VarInt:Integer,VarLong:Long)	
x	Math.HWRD_to_Long(VarLong:Long,VarInt:Integer)	
	Math.INC_Integer(VarInt:Y)	
	Math.INC_Long(VarLong)	
	Math.LN_Single(VarSingle:Y,VarSingle:X)	
	Math.LOG_Single(VarSingle:Y,VarSingle:X)	
x	Math.LWRD_from_Long(VarInt:Integer,VarLong:Long)	
x	Math.LWRD_to_Long(VarLong:Long,VarInt:Integer)	
x	Math.MAX_Integer(VarInt:MAX,VarInt:A,VarInt:B)	
	Math.MEAN_Integer(VarInt:MEAN,VarInt:A,VarInt:B)	
x	Math.MIN_Integer(VarInt:MIN,VarInt:A,VarInt:B)	
x	Math.MOD_Integer(VarInt:Y,VarInt:A,VarInt:B)	Modulo: Stores the remainder on division of A by B to VarInt:Y (10/6=4)
	Math.MOVE_Integer(VarInt,VarInt)	
	Math.MOVE_Single(VarSingle,VarSingle)	
	Math.MUL_Long(VarLong,VarLong,VarLong)	
	Math.MUL_Single(VarSingle,VarSingle,VarSingle)	
	Math.MULS_Integer(VarInt:Res,VarInt:Multiplikand,VarInt:Multiplikator)	

Math.NEG_Integer(VarInt:Y,VarInt:A)	
Math.NEG_Long (VarLong:Y,VarLong:A)	
Math.POWER_Single(VarSingle:Result,VarSingle:x,VarSingle:exponent)	
Math.SIN_Single(VarSingle:Y,VarSingle:Winkel_RAD)	
Math.SM_Get_SP(VarInt:StackPointer)	
Math.SQR_Single(VarSingle:Y,VarSingle:X)	
Math.SQRT_Single(VarSingle:Y,VarSingle:X)	
Math.SUB_Integer(VarInt:Y,VarInt:A,VarInt:B)	
Math.SUB_Long(VarLong,VarLong,VarLong)	
Math.SUB_Single(VarSingle,VarSingle,VarSingle)	
Math.TAN_Single(VarSingle:Y,VarSingle:Winkel_RAD)	

MOVE_LONG(VarLong,VarLong)	
ON_ERROR_GOTO(labelRelative24)	
ON_EVENT_GOTO(labelRelative24)	
ON_EVENT_Touch_GOTO(labelRelative24)	
ReturnFromProject()	wechselt ins aufrufende Projekt zurück

Class: Serial

x	Serial.ReceiveToFile(VarInt:COMx,VarStr:FileName)	
x	Serial.Rx_Char(VarInt:COMx,VarInt:Char)	Receives a Character from COMx, 0xFFFF if no Character present
x	Serial.SendFromFile(VarInt:COMx,VarStr:FileName)	
x	Serial.SetBaudrate(VarInt:COMx,VarInt:Baud_9600)	Sets the Communication Speed of COMx
x	Serial.Tx_Char(VarInt:COMx,VarInt:Char)	Transmits a Character over COMx
x	Serial.Tx_CRLF(VarInt:COMx)	Transmits a CR and a LF Character over COMx
x	Serial.Tx_NUL(VarInt:COMx)	Transmits a NUL Character over COMx
x	Serial.Tx_String(VarInt:COMx,VarStr:String)	Transmits a String over COMx

Class: Sound

	Sound.InstallLocal(labelRelative24:Event)	
	Sound.PlaySamples(VarInt:Sound_MODE_STEREO,VarStr:File,VarLong:StartOffset,VarLong:NoOfSamples,VarInt:Loops)	
	Sound.SetBitsPerSample(VarInt:BitsPerSample)	
	Sound.SetSamplingRate(VarInt:SamplingRate_11025)	

Class: Store

	Store.Data_to_ARRAY_Byte(labelRelative24,VarInt:Index,VarInt:Value)	
	Store.Data_to_ARRAY_Integer(labelRelative24,VarInt:Index,VarInt:Value)	
	Store.Data_to_ARRAY_Long(labelRelative24,VarInt:Index,VarLong:Value)	
	Store.Data_to_ARRAY_Single(labelRelative24,VarInt:Index,VarSingle:Value)	

Class: Str (String)

x	Str.Add_CRLF(VarStr:DestString)	Adds CR LF at the end of
---	---------------------------------	--------------------------

		DestString
	Str.Add_ProjectName(VarStr:DestString)	Adds the ProjectName (4 Characters) at the end of DestString
x	Str.AddChar(VarStr:DestString,VarInt:Character)	Adds a Character at the end of DestString
x	Str.BufferLength(VarInt:Length,VarStr:SourceString)	Determines the maximum length of SourceString
	Str.Checksum(VarStr:String,VarInt:ChecksumFunction,VarInt:Checksum)	
x	Str.Clear(VarStr:DestString)	löscht den Deststring
x	Str.Compare(VarInt:Ergebnis,VarStr:Vergleichsstring_1,VarStr:Vergleichsstring_2)	
x	Str.Concat(VarStr:DestString,VarStr:SourceString)	
	Str.ConcatMove(VarStr:DestString,VarStr:SourceString1,VarStr:SourceString2)	
x	Str.Copy(VarStr:DestString,VarStr:SourceString)	
x	Str.CopySubstring_to_LineEnd(VarStr:Dest,VarStr:Quelle,VarInt:Pos,VarInt:NewPos)	
	Str.CopySubstringWord(VarStr:Dest,VarStr:SourceString,VarInt:Pos)	
x	Str.Cvt_ByteHex(VarStr:DestString,VarInt:LSB)	
x	Str.Cvt_Integer(VarStr:DestString,VarInt:Zahl,VarInt:Feldlaenge)	
	Str.Cvt_IntegerDeci(VarStr:DestString,VarInt:Zahl,VarInt:Vorkomma,VarInt:Nachkomma)	
x	Str.Cvt_Long(VarStr:DestString,VarLong:Zahl,VarInt:Stellen)	
x	Str.Cvt_LongDeci(VarStr:DestString,VarLong:Zahl,VarInt:Vorkomma,VarInt:Nachkomma)	
x	Str.Cvt_LongHex(VarStr:DestString,VarLong)	
	Str.Cvt_Single(VarStr:DestString,VarSingle:Zahl,VarInt:Vorkomma,VarInt:Nachkomma)	
	Str.Cvt_Uinteger(VarStr:DestString,VarInt:Zahl,VarInt:Feldlaenge)	
	Str.Cvt_UintegerDeci(VarStr:DestString,VarInt:Zahl,VarInt:Vorkomma,VarInt:Nachkomma)	
x	Str.Cvt_WordHex(VarStr:DestString,VarInt)	
x	Str.Date(VarStr:DestString,VarInt:FormatDate)	
x	Str.FillUpWithChar(VarStr:DestString,VarInt:Character)	
x	Str.Find(VarStr:VergleichString,VarInt:StartPosition,VarStr:SuchString,VarInt:FundPosition)	
	Str.GetChar_at_Position(VarInt:Char,VarInt:Position,VarStr:SourceString)	Copies a character at a given position of SourceString to an integer variable
x	Str.GetPosition_by_Char(VarInt:FoundPosition,VarInt:Char,VarInt:StartPosition,VarStr:InputString)	Detects the position of a given character and start searching at a given position
	Str.Insert_n_Chars_at_Position(VarInt:Char,VarInt:Position,VarInt:NumberOfChars,VarStr:String)	
	Str.InsertChar_at_Position(VarInt:Char,VarInt:Position,VarStr:String)	
	Str.Left(VarStr:DestString,VarStr:InputString,VarInt:NoOfChars)	
x	Str.Length(VarInt:Length,VarStr:QuellString)	Detects the number of characters of SourceString
x	Str.LowerCase(VarStr:DestString)	Converts the characters of DestString to lower-case characters
x	Str.Match(VarStr:CompareString,VarInt:StartPosition,VarStr:SearchString)	
	Str.Mid(VarStr:DestString,VarStr:InputString,VarInt:Position,VarInt:NoOfChars)	
	Str.MovePosition_to_LineNo(VarStr:Quelle,VarInt:LineNo,VarInt:NewPos)	
	Str.MovePosition_to_NextLine(VarStr:Quelle,VarInt:LineNo,VarInt:NewPos)	

	Str.PutChar_at_Position(VarInt:Char,VarInt:Position,VarStr:String)	Inserts a character at a given position of the string
	Str.RemoveChar_at_Position(VarInt:RemovedChar,VarStr:String,VarInt:Position)	Removes a character at a given position of the string
x	Str.RemoveLastChar(VarInt:RemovedChar,VarStr:String)	
	Str.RemoveWhiteSpace_at_Position(VarInt:NonWhiteSpaceChar,VarStr:String,VarInt:Position)	
	Str.Right(VarStr:DestString,VarStr:InputString,VarInt:NoOfChars)	
x	Str.SkipBlank(VarStr:String,VarInt:StartPosition,VarInt:FindPosition)	
x	Str.SpaceInString(VarInt:Space,VarStr:SourceString)	Detects how many characters still can be added to the string
x	Str.Time(VarStr:DestString,VarInt:FormatTime)	
x	Str.UpperCase(VarStr:DestString)	Converts the characters of DestString to upper-case characters
	Str.WriteSubString(VarStr:DestString,VarInt:Position,VarStr:SourceString)	

Class: System

	System.Get_ms_Counter(VarLong)	
	System.Reset()	Resets the system
	System.Reset_ms_Counter()	

Class: Time

x	Time.DayOfWeek()	
x	Time.Get()	
x	Time.Set()	
x	Time.SetDate()	
x	Time.SetTime()	

Class: Timer

x	Timer.Continue(VarInt :Timer)	
x	Timer.DeInstall(VarInt:Timer)	
x	Timer.Get_ExpiredCounter(VarInt:Timer,VarInt:TimerCounter)	
x	Timer.Get_TimerCounter(VarInt:Timer,VarInt:TimerCounter)	
x	Timer.Init()	
x	Timer.InstallGlobal(VarInt :Timer, labelAbsolute :Event)	
x	Timer.InstallLocal(VarInt :Timer, labelRelative24 :Event)	
x	Timer.Kill(VarInt :Timer)	
x	Timer.Load(VarInt:Timer,VarInt:Zeit)	
x	Timer.Load_ExpiredCounter(VarInt:Timer,VarInt:Wert)	
x	Timer.Reload(VarInt:Timer)	
x	Timer.StartContinuous(VarInt:Timer)	
x	Timer.StartSingle(VarInt:Timer)	
x	Timer.Stop(VarInt:Timer)	
x	Timer.SUB_ExpiredCounter(VarInt:Timer,VarInt:Wert)	
x	Timer.TableDisable()	
x	Timer.TableEnable()	

Class: Transfer

x	Transfer.HotSpotGeometry()	
	Transfer.HotSpotOffset()	
	Transfer.Long_To_Single(VarLong:Long,VarSingle:Single)	

	Transfer.Offset_from_Pos_X1Y1()	
	Transfer.Single_To_Long(VarSingle:Single,VarLong:Long)	
	Transfer.To_HotSpotInfo()	

Class: Value

x	Value.ByteHex(VarInt:Number,VarStr:InputString,VarInt:Position)	
	Value.HighColor(VarInt:Color,VarStr:InputString,VarInt:Position)	
x	Value.Integer(VarInt:Number,VarStr:InputString,VarInt:Position)	
x	Value.Long(VarLong:Number,VarStr:InputString,VarInt:Position)	
	Value.LongDeci(VarLong:Number,VarStr:InputString,VarInt:Position,VarInt:Nachkomma)	
x	Value.LongHex(VarLong:Number,VarStr:InputString,VarInt:Position)	
x	Value.UInteger(VarInt:Number,VarStr:InputString,VarInt:Position)	
x	Value.WordHex(VarInt:Number,VarStr:InputString,VarInt:Position)	

■ Support

Have you got questions concerning programming language eigerScript or products of eigergraphics? We hope you'll find the answer at www.eigergraphics.com. Otherwise don't hesitate contacting us:

Phone: +41 41 754 50 10

Email: eiger@s-tec.ch