**VERSION: 3.12.13193 [20/JAN/2023]**

**Uses VPixx Device Server**

**% Setup functions:**

isReady = Datapixx('Open');

isReady = Datapixx('OpenIP', ip);

selectedDevice = Datapixx('SelectDevice', deviceType=-1, deviceName);

isReady = Datapixx('IsReady');

isDatapixx = Datapixx('IsDatapixx');

isDatapixx2 = Datapixx('IsDatapixx2');

isDatapixx3 = Datapixx('IsDatapixx3');

isViewpixx = Datapixx('IsViewpixx');

isViewpixx3D = Datapixx('IsViewpixx3D');

isPropixxCtrl = Datapixx('IsPropixxCtrl');

isPropixx = Datapixx('IsPropixx');

isPropixx = Datapixx('IsTrackpixx');

Datapixx('Close');

**% Global system information:**

ramSize = Datapixx('GetRamSize');

firmwareRev = Datapixx('GetFirmwareRev');

time = Datapixx('GetTime');

Datapixx('SetMarker');

marker = Datapixx('GetMarker');

supplyVoltage = Datapixx('GetSupplyVoltage');

supplyCurrent = Datapixx('GetSupplyCurrent');

is5VFault = Datapixx('Is5VFault');

tempCelcius = Datapixx('GetTempCelcius');

tempFarenheit = Datapixx('GetTempFarenheit');

**% DAC (Digital to Analog Converter) subsystem:**

dacNumChannels = Datapixx('GetDacNumChannels');

dacRanges = Datapixx('GetDacRanges');

Datapixx('SetDacVoltages', channelVoltagePairs);

dacVoltages = Datapixx('GetDacVoltages');

[nextBufferAddress, underflow, overflow] = Datapixx('WriteDacBuffer', bufferData,...

*bufferAddress*=0, *channelList*=0:numChannels-1);

Datapixx('SetDacSchedule', scheduleOnset, scheduleRate, maxScheduleFrames, ...

*channelList*=0, *bufferBaseAddress*=0, *numBufferFrames*=maxScheduleFrames);

Datapixx('StartDacSchedule');

Datapixx('StopDacSchedule');

status = Datapixx('GetDacStatus');

**% ADC (Analog to Digital Converter) subsystem:**

adcNumChannels = Datapixx('GetAdcNumChannels');

adcRanges = Datapixx('GetAdcRanges');

adcVoltages = Datapixx('GetAdcVoltages');

Datapixx('EnableDacAdcLoopback');

Datapixx('DisableDacAdcLoopback');

Datapixx('EnableAdcFreeRunning');

Datapixx('DisableAdcFreeRunning');

Datapixx('SetAdcSchedule', scheduleOnset, scheduleRate, maxScheduleFrames,...

*channelList*=0, *bufferBaseAddress*=4e6, *numBufferFrames*=maxScheduleFrames);

Datapixx('StartAdcSchedule');

Datapixx('StopAdcSchedule');

[bufferData, bufferTimetags, underflow, overflow] = Datapixx('ReadAdcBuffer', ...

    numFrames, bufferAddress);

status = Datapixx('GetAdcStatus');

**% DOUT (Digital Output) subsystem:**

doutNumBits = Datapixx('GetDoutNumBits');

Datapixx('SetDoutValues', bitValues, bitMask = hex2dec('00FFFFFF'))

doutValues = Datapixx('GetDoutValues');

Datapixx('EnableDoutButtonSchedules', mode = 0);

Datapixx('DisableDoutButtonSchedules');

Datapixx('EnableDoutBacklightPulse');

Datapixx('DisableDoutBacklightPulse');

Datapixx('EnableDoutBlink')

Datapixx('DisableDoutBlink')

[nextBufferAddress, underflow, overflow] = Datapixx('WriteDoutBuffer',...

    bufferData, bufferAddress=8e6);

Datapixx('SetDoutSchedule', scheduleOnset, scheduleRate, maxScheduleFrames,...

*bufferBaseAddress*=8e6, *numBufferFrames*=maxScheduleFrames);

Datapixx('StartDoutSchedule');

Datapixx('StopDoutSchedule');

Datapixx('EnablePixelMode', mode = 0);

Datapixx('DisablePixelMode');

Datapixx('EnableVsyncMode');

Datapixx('DisableVsyncMode');

status = Datapixx('GetDoutStatus');

**% DIN (Digital Input) subsystem:**

dinNumBits = Datapixx('GetDinNumBits');

dinValues = Datapixx('GetDinValues');

Datapixx('SetDinDataDirection', directionMask);

Datapixx('SetDinDataOut', dataOut);

doutValues = Datapixx('GetDinDataOut');

Datapixx('SetDinDataOutStrength', strength);

Datapixx('EnableDinDebounce');

Datapixx('DisableDinDebounce');

Datapixx('EnableDoutDinLoopback');

Datapixx('DisableDoutDinLoopback');

Datapixx('SetDinLog', *bufferBaseAddress*=12e6, *numBufferFrames*=1000);

Datapixx('StartDinLog');

Datapixx('StopDinLog');

[logData, logTimetags, underflow] = Datapixx('ReadDinLog', numFrames);

status = Datapixx('GetDinStatus');

**% TOUCHPixx (touch panel) subsystem:**

Datapixx('EnableTouchpixx', touchPanelMode=0);

Datapixx('DisableTouchpixx');

coordinates = Datapixx('GetTouchpixxCoordinates');

Datapixx('SetTouchpixxStabilizeDuration', duration);

Datapixx('SetTouchpixxLog', bufferBaseAddress=12e6, numBufferFrames=1000);

Datapixx('StartTouchpixxLog');

Datapixx('StopTouchpixxLog');

[logCoords, logTimetags, underflow] = Datapixx('ReadTouchpixxLog', numFrames);

Datapixx('EnableTouchpixxLogContinuousMode');

Datapixx('DisableTouchpixxLogContinuousMode');

status = Datapixx('GetTouchpixxStatus');

**% Video subsystem:**

Datapixx('SetVideoMode', mode=0);

Datapixx('SetVideoGreyscaleMode', mode=1);

Datapixx('SetVideoClut', clut);

Datapixx('SetVideoClutTransparencyColor', color);

Datapixx('EnableVideoClutTransparencyColorMode');

Datapixx('DisableVideoClutTransparencyColorMode');

Datapixx('SetVideoHorizontalSplit', mode(0=MIRROR,1=SPLIT,2=AUTO));

Datapixx('SetVideoVerticalStereo', mode(0=NO\_STEREO,1=STEREO,2=AUTO));

Datapixx('SetVideoStereoEye', eye(1=Left,0=Right));

Datapixx('EnableVideoStereoBlueline');

Datapixx('DisableVideoStereoBlueline');

Datapixx('SetVideoStereoVesaWaveform', waveform);

Datapixx('SetVideoStereoVesaPhase', phase);

Datapixx('EnableVideoHorizontalOverlay');

Datapixx('DisableVideoHorizontalOverlay');

Datapixx('SetVideoHorizontalOverlayBounds', boundsRect);

Datapixx('SetVideoHorizontalOverlayAlpha', alphaTable);

Datapixx('SetVideoPixelSyncLine', rasterLine, *singleLine*=1, *blankLine*=1);

Datapixx('EnableVideoScanningBacklight');

Datapixx('DisableVideoScanningBacklight');

Datapixx('EnableVideoRescanWarning');

Datapixx('DisableVideoRescanWarning');

Datapixx('SetVideoBacklightIntensity', intensity);

Datapixx('EnableVideoLcd3D60Hz');

Datapixx('DisableVideoLcd3D60Hz');

pixels = Datapixx('GetVideoLine', *nPixels*=HORIZONTAL\_RESOLUTION);

status = Datapixx('GetVideoStatus');

Datapixx('SetVideoConsoleDisplay', *presetDisposition*=0);

**% PROPixx-specific routines:**

Datapixx('SetPropixxDlpSequenceProgram', *program*=0);

Datapixx('EnablePropixxCeilingMount');

Datapixx('DisablePropixxCeilingMount');

Datapixx('EnablePropixxRearProjection');

Datapixx('DisablePropixxRearProjection');

Datapixx('SetPropixx3DCrosstalk', crosstalk);

Datapixx('SetPropixx3DCrosstalkLR', crosstalk);

Datapixx('SetPropixx3DCrosstalkRL', crosstalk);

Datapixx('EnablePropixxLampLed');

Datapixx('DisablePropixxLampLed');

Datapixx('EnableHotspotCorrection');

Datapixx('DisableHotspotCorrection');

Datapixx('SetPropixxAwake');

Datapixx('SetPropixxSleep');

isAwake = Datapixx('IsPropixxAwake');

Datapixx('SetPropixxLedMask', *mask*=0);

Datapixx('EnablePropixxQuad4x3d');

Datapixx('DisablePropixxQuad4x3d');

Datapixx('SetGrayLEDCurrents', *index*=0);

Datapixx('SetCustomCalibrationCurrents', *index*=0);

Datapixx('SetPropixxLedIntensity', *mode*=0);

**% PROPixx T-Scope Mode routines**

Datapixx('EnablePropixxTScope');

Datapixx('DisablePropixxTScope');

Datapixx('EnablePropixxTScopePrepRequest');

Datapixx('DisablePropixxTScopePrepRequest');

Datapixx('WritePropixxTScopePages', pageIndex, pageData);

Datapixx('SetPropixxTScopeSchedule', scheduleOnset, scheduleRate, ...

    maxScheduleFrames, *startPage*=0, *nPages*=maxScheduleFrames]);

Datapixx('StartPropixxTScopeSchedule');

Datapixx('StopPropixxTScopeSchedule');

Datapixx('SetPropixxTScopeMode', *mode*=0);

Datapixx('SetPropixxTScopeProgramAddress', *addr*=0);

Datapixx('SetPropixxTScopeProgramOffsetPage', offset);

Datapixx('SetPropixxTScopeProgram', program);

**% TRACKPixx (any kind) Functions:**

Datapixx('GetEyeDuringCalibration', xScreen, yScreen);

[xRawRight, yRawRight, xRawLeft, yRawLeft] = Datapixx( ...

    'GetEyeDuringCalibrationRaw', xScreen, yScreen);

Datapixx('FinishCalibration');

calibrations\_coeff = Datapixx('GetCalibrationCoeff');

[xScreenRight, yScreenRight, xScreenLeft, yScreenLeft,...

xRawRight, yRawRight, xRawLeft, yRawLeft, timetag] = Datapixx('GetEyePosition');

convertedArray = Datapixx('ConvertCoordSysToCartesian', ...

    sourceArray, offsetX, scaleX, offsetY, scaleY);

convertedArray = Datapixx('ConvertCoordSysToCustom', ...

    sourceArray, offsetX, scaleX, offsetY, scaleY);

**% TRACKPixx3 only Functions:**

Datapixx('SaveCalibration');

Datapixx('LoadCalibration');

Datapixx('ClearCalibration');

image = Datapixx('GetEyeImage');

Datapixx('SetLedIntensity', ledIntensity);

ledIntensity = Datapixx('GetLedIntensity');

Datapixx('SetExpectedIrisSizeInPixels', IrisSize);

expectedIrisSize = Datapixx('GetExpectedIrisSizeInPixels');

pupilSize = Datapixx('GetPupilSizeSimple');

[ppLeftMajor, ppLeftMinor, ppRightMajor, ppRightMinor] = Datapixx('GetPupilSize');

[ppLeftX, ppLeftY, ppRightX, ppRightY] = Datapixx('GetPupilCoordinatesInPixels');

[CRLeftX, CRLeftY, CRRightX, CRRightY] = Datapixx('GetCRCoordinatesInPixels');

Datapixx('SetupTPxSchedule', *bufferbaseAddress*=12e6, *numberOfEyeData*=60000);

Datapixx('StopTPxSchedule');

Datapixx('StartTPxSchedule');

[bufferData, underflow, overflow] = Datapixx('ReadTPxData', numFrames);

status = Datapixx('GetTPxStatus');

Datapixx('ShowOverlay');

Datapixx('HideOverlay');

Datapixx('SetTPxSleep');

Datapixx('SetTPxAwake');

Datapixx('EnableSearchLimits');

Datapixx('DisableSearchLimits');

Datapixx('ClearSearchLimits');

Datapixx('SetSearchLimits', leftEye, rightEye);

[leftEye, rightEye] = Datapixx('GetSearchLimits');

Datapixx('EnableTPxAnalogOut', *DAC0*=0, *DAC1*=0, *DAC2*=0, *DAC3*=0);

Datapixx('DisableTPxAnalogOut');

Datapixx('PpSizeCalGetData');

Datapixx('PpSizeCalGetDataComplete');

Datapixx('PpSizeCalLinearRegression');

Datapixx('PpSizeCalClear');

Datapixx('PpSizeCalSet');

[slope\_r\_x, slope\_r\_y, slope\_l\_x, slope\_l\_y] = Datapixx('PpSizeCalGet');

fov\_h = Datapixx('GetHorizontalFOV');

fov\_v = Datapixx('GetVerticalFOV');

pxSize = Datapixx('GetPixelSize');

pxDensity = Datapixx('GetPixelDensity');

Datapixx('SetLens', lens);

lens = Datapixx('GetLens');

Datapixx('SetDistance', distance);

distance = Datapixx('GetDistance');

[leftFixationFlag, rightFixationFlag] = Datapixx('IsSubjectFixating');

[leftSaccadeFlag, rightSaccadeFlag] = Datapixx('IsSubjectMakingSaccade');

Datapixx('SetFixationThresholds', *maxSpeed*=2500, *minNumberOfConsecutiveSamples*=25);

Datapixx('SetSaccadeThresholds', *minSpeed*=10000, *minNumberOfConsecutiveSamples*=10);

Datapixx('SetTrackingMode', mode);

distance = Datapixx('GetTrackingMode');

Datapixx('SetTrackingSpecies', Species);

distance = Datapixx('GetTrackingSpecies');

**% Reading and writing register cache:**

Datapixx('RegWr');

Datapixx('RegWrRd');

Datapixx('RegWrVideoSync');

Datapixx('RegWrRdVideoSync');

Datapixx('RegWrPixelSync', pixelSequence, *timeout*=255);

isTimeout = Datapixx('RegWrRdPixelSync', pixelSequence, *timeout*=255);

**% Miscellaneous Routines:**

Datapixx('StopAllSchedules');

error = Datapixx('GetError');

Datapixx('ClearError');

Datapixx('Reset');

Datapixx('ResetAll');