ShadowSense Device Interface

ShadowSense is available as a nugget package, this is required in order to use the api. In the package for HID there is one interface described below:

```
public interface IShadowSenseDevice : IDisposable
{
    event EventHandler<InsertedEvent> Inserted;
    event EventHandler<RemovedEvent> Removed;
    event EventHandler<SettingsProgressEvent> SettingsProgressChanged;

    event EventHandler<DeviceConfigurationEvent> DeviceConfigurationChanged;

    event EventHandler<ShadowSenseTouchEvent> TouchDown;
    event EventHandler<ShadowSenseTouchEvent> TouchMove;
    event EventHandler<ShadowSenseTouchEvent> TouchUp;


    bool IsOpen { get; }
    bool IsConnected { get; }
    string DevicePath { get; }
    DeviceConfiguration DeviceConfiguration { get; }
    string Name { get; }
    Version Version { get; }
    string Serial { get; }

    void CloseDevice();

    bool EnterDFU();

    /// <summary>
    /// Reads all settings from device.
    /// </summary>
    /// <returns>true or false success of call.</returns>
    bool LoadSettings(ref DeviceSettings settings);

    /// <summary>
    /// Writes all settings to device.
    /// </summary>
    /// <returns>true or false success of call.</returns>
    bool SaveSettings(DeviceSettings settings);

    #region Calibration Settings

    /// <summary>
    /// Reads usb boot delay from device.
    /// </summary>
    /// <returns>true or false success of call.</returns>
    bool GetUsbBootDelay(ref byte delay);
    /// <summary>
    /// Reads new touch delay from device.
    /// </summary>
    /// <returns>true or false success of call.</returns>
    bool GetNewTouchDelay(ref byte delay);
```

```
/// <summary>
/// Reads separation threshold from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetSeparationThreshold(ref int data);
/// <summary>
/// Reads maximum shadow from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetMaxShadow(ref byte data);
/// <summary>
/// Reads minimum shadow from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetMinShadow(ref byte data);
/// <summary>
/// Reads filter size from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetFilterSize(ref byte data);
/// <summary>
/// Reads calibration from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetCalibrationPeriod(ref byte data);
/// <summary>
/// Reads recovery speed from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetRecoverySpeed(ref byte data);
/// <summary>
/// Reads filter depth from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetFilterDepth(ref byte data);


/// <summary>
/// Writes usb boot delay to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetUsbBootDelay(byte delay);
/// <summary>
/// Writes new touch delay to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetNewTouchDelay(byte delay);
/// <summary>
/// Writes separation threshold to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetSeparationThreshold(int data);
/// <summary>
/// Writes maximum shadow to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetMaxShadow(byte data);
/// <summary>
/// Writes minimum shadow to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetMinShadow(byte data);
/// <summary>
/// Writes filter size to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetFilterSize(byte data);
/// <summary>
/// Writes calibration period to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetCalibrationPeriod(byte data);
/// <summary>
/// Writes recovery speed to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetRecoverySpeed(byte data);
/// <summary>
/// Writes filter depth to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetFilterDepth(byte data);


#endregion

#region Calibration Ex Settings

/// <summary>
/// Reads touch reject enable from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetTouchRejectEnable(ref byte data);
/// <summary>
/// Reads touch reject maximum area from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetTouchRejectMaximumArea(ref int data);
/// <summary>
/// Reads touch reject minimum area from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetTouchRejectMinimumArea(ref int data);
/// <summary>
/// Reads active mode led brightness from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetActiveModeLedBrightness(ref byte data);
/// <summary>
/// Reads active mode frame time from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetActiveModeFrameTime(ref byte data);
/// <summary>
/// Reads idle mode brightness from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetIdleModeLedBrightness(ref byte data);
/// <summary>
/// Reads touch reject enable from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetIdleModeFrameTime(ref int data);
/// <summary>
/// Reads idle mode time until idle from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetIdleModeTimeUntilIdle(ref int data);


/// <summary>
/// Writes touch reject enable to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetTouchRejectEnable(byte data);
/// <summary>
/// Writes touch reject maximum area to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetTouchRejectMaximumArea(int data);
/// <summary>
/// Writes touch reject minimum area to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetTouchRejectMinimumArea(int data);
/// <summary>
/// Writes active mode led brightness to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetActiveModeLedBrightness(byte data);
/// <summary>
/// Writes active mode frame time to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetActiveModeFrameTime(byte data);
/// <summary>
/// Writes idle mode brightness to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetIdleModeLedBrightness(byte data);
/// <summary>
/// Writes touch reject enable to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetIdleModeFrameTime(int data);
/// <summary>
/// Writes idle mode time until idle to device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool SetIdleModeTimeUntilIdle(int data);

#endregion

#region Modes
```

```
/// <summary>
/// Reads mouse mode from device.
/// </summary>
/// <returns>true or false success of call.</returns>
bool GetMouseMode(ref byte data);
bool GetWindowsTouchPipeEnable(ref bool data);
bool GetWindowsTouchDataEnable(ref bool data);
bool GetShadowSenseTouchPipeEnable(ref bool data);
bool GetShadowSenseTouchDataEnable(ref bool data);
bool GetMousePipeEnable(ref bool data);
bool GetMouseDataEnable(ref bool data);
bool GetMouseRightClickEnable(ref bool data);
bool GetMouseMoveTrackModeEnable(ref bool data);
bool GetMouseRightClickSensitivity(ref byte data);
bool GetMouseDeadband(ref byte data);

bool SetMouseMode(MouseMode data);
bool SetWindowsTouchPipeEnable(bool data);
bool SetWindowsTouchDataEnable(bool data);
bool SetShadowSenseTouchPipeEnable(bool data);
bool SetShadowSenseTouchDataEnable(bool data);
bool SetMousePipeEnable(bool data);
bool SetMouseDataEnable(bool data);
bool SetMouseRightClickEnable(bool data);
bool SetMouseMoveTrackModeEnable(bool data);
bool SetMouseRightClickSensitivity(byte data);
bool SetMouseDeadband(byte data);


#endregion

#region Stylus

bool GetStylusEnable(ref bool data);
bool GetEraserEnable(ref bool data);
bool GetStylusMaximumArea(ref int data);
bool GetStylusMinimumArea(ref int data);
bool GetEraserMaximumArea(ref int data);
bool GetEraserMinimumArea(ref int data);

bool SetStylusEnable(bool data);
bool SetEraserEnable(bool data);
bool SetStylusMaximumArea(int data);
bool SetStylusMinimumArea(int data);
bool SetEraserMaximumArea(int data);
bool SetEraserMinimumArea(int data);


#endregion

#region Stylus Shadows
bool GetStylusShadowMaximumArea(ref int data);
bool GetStylusShadowMinimumArea(ref int data);
bool GetEraserShadowMaximumArea(ref int data);
bool GetEraserShadowMinimumArea(ref int data);
bool GetStylusPalmRejectionRadius(ref int data);

bool SetStylusShadowMaximumArea(int data);
```

```
    bool SetStylusShadowMinimumArea(int data);
    bool SetEraserShadowMaximumArea(int data);
    bool SetEraserShadowMinimumArea(int data);
    bool SetStylusPalmRejectionRadius(int data);
    #endregion

    #region Water
    bool GetWaterRejectMode(ref byte data);
    bool GetWaterRejectAmount(ref byte data);
    bool GetWaterShadowMaximumArea(ref int data);
    bool GetWaterShadowMinimumArea(ref int data);

    bool SetWaterRejectMode(WaterRejectMode data);
    bool SetWaterRejectAmount(byte data);
    bool SetWaterShadowMaximumArea(int data);
    bool SetWaterShadowMinimumArea(int data);
    #endregion

    #region Screen Mask
    bool GetTouchConfidence(ref byte data);
    bool GetScreenMask(ref ScreenMask data);

    bool SetTouchConfidence(byte data);
    bool SetScreenMask(ScreenMask data);

    #endregion

    #region Transform
    bool GetTransformMode(ref bool data);
    bool GetTransform(ref object data, ref bool mode);
    bool GetMacTransformEnable(ref bool data);
    bool GetScreenOrientation(ref ScreenOrientation data);
    bool SetTransformMode(bool data);
    bool SetTransform(Matrix3D data);
    bool SetAltTransform(AltTransform data);
    bool SetMacTransformEnable(bool data);
    bool SetScreenOrientation(ScreenOrientation data);
    #endregion

    #region Regions
    bool GetRegionsEnable(ref byte data);
    bool GetRegionMask(ref ScreenMask data, Regions id);
    bool SetRegionsEnable(byte data);
    bool SetRegionMask(ScreenMask data, Regions id);

    #endregion
}
```