**Decoding SDK (Android) – Interface Description**

1. Files:

BarcodeReader2DTG\libs\TwoDtgDecodeLib2-release.aar – archaive, containing decoders.

BarcodeReader2DTG\libs\CameraLib-release.aar - archaive, containing libraries for working with camera and/or Imager (Honeywell N56XX – Dolphin 70e, 75e, CT50; Nautiz X4, Newland MT-90)

1. Packages:

decoding.twodtg.com.twodtgdecodelib2 – contains decoders and auxiliary classes

decoding.twodtg.com.cameralib – contains interface for camera or Imager.

1. Classes.

* 1. BarcodeDecoder

Class for all decoders. Describes their interface.

**Important!**

The following permissions are required for BarcodeDecoder:

* Android permissions for Decoder itself:

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>  
<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.READ\_PHONE\_STATE"/> (for Android 8 and higer)

* Android permission for image capture by a smartphone/tablet PC camera and its subsequent decoding:

<**uses-permission android:name="android.permission.CAMERA"**/>

* Honeywell permission for image capture by an imager of a Honeywell industrial mobile computer and its subsequent decoding:

<**uses-permission android:name="com.honeywell.decode.permission.DECODE"**/>

* Android permissions for license on-line activation/deactivation:

<**uses-permission android:name="android.permission.INTERNET"**/>

Contains methods as follows:

3.1.1 public static boolean isStandardCamera()

Result:

Returns true if smartphone camera is used for image capture.

3.1.2 public void setTriggerListener(TriggerListener listener)

Parameters:

TriggerListener listener – reciprocal call object.

Sets reciprocal call object when trigger is pressed/released (for devices equipped with a trigger).

3.1.3 public void Destroy()

Releases all resources occupied by the library.

3.1.4 public boolean DecodeBitmap(Bitmap bmp)

Parameters:

Bitmap bmp - image

Result:

Returns true if decoding is successful.

Decodes image presented in Bitmap format.

3.1.5 public boolean DecodeGrayScaleByteArray(byte[] image, int width, int height)

Parameters:

byte[] image - grayscale image array

int width - image width

int height - image height

Result:

Returns true if decoding is successful.

Decodes image presented in the form of grayscale array.

3.1.6 public boolean DecodeCamera(Context context, SurfaceHolder holder, DecodeOkDelegate dlgVal, CameraImageSize CameraSize)

Parameters:

Context context - application context

SurfaceHolder holder - surface for mapping video from a camera

DecodeOkDelegate dlgVal - reciprocal call object if decoding is successful

CameraImageSize CameraSize – camera resolution selection (not applicable for the Imager)

Launches asynchronous process of image capture (from camera or Imager) and decoding.

Calls DecodeOK() of dlgVal object if decoding is successful.

3.1.7 public void StopDecodeCamera()

Stops the process of interaction with the camera.

3.1.8 public Point[] GetBarcodeRect()

Return coordinates of barcode corners.

3.1.9 public String GetDecodeResult(Charset charset)

Parameters:

Charset charset – charset for barcode data

Result:

Barcode content

Returns barcode content.

3.1.10 public void setVideoView(View v)

Parameters:

View v - View surface for mapping video from a camera or Imager.

Sets View surface for mapping video from a camera or Imager.

3.1.11 public String CodeName()

Result:

Barcode type.

Returns barcode type.

3.1.12 **public static** BarcodeDecoder CreateDecoderForCamera(Context context, **int** barType)

Creates a decoder and connects it to a camera.

Only one decoder allowed to be present at any given moment.

Parameters:

Context context – application context

**int** barType – barcode type(obtained by combining constant from BarcodeType)

Result:

Decoder object.

3.1.13 **public static** BarcodeDecoder CreateDecoderForImages(Context context, **int** barType)

Creates a decoder.

Only one decoder allowed to be present at any given moment.

Parameters:

Context context – application context

**int** barType – barcode type(obtained by combining constant from BarcodeType)

Result:

Decoder object.

3.1.12 public byte[] GetDecodeBytes()

Result:

RAW decoded data.

3.1.13 public boolean AddUserId(String userId)

Uses in DMPS version of the SDK.

Add autentication key.

Parameters:

String userId – new UserId

Result:

Return true if new UserId successfully added to the decoder.

3.1.14. public void ClearUserIds()

Uses in DMPS version of the SDK.

Remove all UserId from decoder.

3.1.15. public int GetAutenticationResult()

Uses in DMPS version of the SDK.

Returns the result of authentication.

Result:

The return value lies within the range of values contained in the enum DataMatrixAutenticationResult.

DataMatrixAutenticationResult:

NotUse - authentication was not performed.

Failed – authentication FAILED.

Authenticated – authentication successful.

Decryption – decryption successful.

Authenticated\_Failed\_Decryption\_OK - decryption successful and authentication is FAILED.

Both - decryption and authentication is successful.

3.2. BarcodeDecoderLibrary- Library activation and other.

Contains the following methods:

3.2.1 public int EvaluationDayCount()

Result:

The number of remaining days of trial or days left for 1 year License.

.

3.2.2 public **boolean** isActivated(Activity context)

Parameters:

Context context – application context

Result:

Returns true if the library is activated.

**3.2.3 public boolean** OnlineActivation(Activity context, **int** licenseId, String psw)

Activates the library on the 2dtg site over the Internet.

**Parameters:**

Activity context – activity

**int** licenseId – license ID

String psw – password

Result:

Returns true if the library is activated.

3.2.4 **public boolean** ManualActivation(Activity context, **int** activationCode1, **int** activationCode2, **int** userCode1, **int** userCode2)

Manual activation of the library.

Parameters:

Activity context – activity

**int** activationCode1 – activation code1 from 2dtg site

**int** activationCode2 – activation code2 from 2dtg site

**int** userCode1 – user code1. Must be pre-generated function UserCode1

**int** userCode2 – user code2. Must be pre-generated function UserCode2

Result:

Returns true if the library is activated.

3.2.5 **public** String UserCode1(Activity context)

Parameters:

Activity context – activity

Result:

Returns user code1.

3.2.6 **public** String UserCode2(Activity context)

Parameters:

Activity context – activity

Result:

Returns user code2.

**3.2.7 public boolean** OnlineDeactivation(Activity context, **int** licenseId, String psw)

Deactivates the library on the 2dtg site over the Internet.

**Parameters:**

Activity context – activity

**int** licenseId – license ID

String psw – password

3.2.8 public boolean DMProtectionSupported()

Result:

Return true if DMPS supported.

3.3 BarcodeDecoderOptions– Contains options the decoder.

1. Interfaces

4.1 DecodeOkDelegate – reciprocal call interface in case of successful decoding

Contains one method:

public void DecodeOK()

Called by the process of asynchronous decoding if decoding is successful

4.2 TriggerListener - reciprocal call interface when the trigger is pressed/released

Contains one method:

boolean TriggerEvent(boolean pressed)

Parameter:

boolean pressed - trigger is pressed/released

Called when trigger is pressed/released.