00A0 $\exists$ $\forall$ $\subseteq$ 2713x 27FA$\Longleftrightarrow$ $\surd$ $\sqrt[3]{}$ $\oplus$ $\otimes$

mynteye_d_android_sdk Documentation

robinge

Oct 11, 2019

Contents:

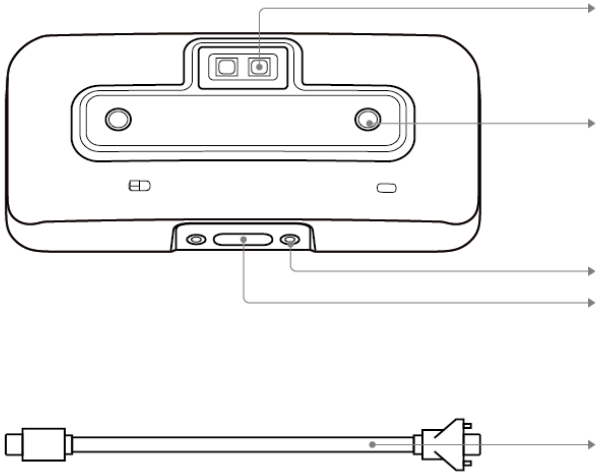
CHAPTER 1

MYNT® EYE 1200

1.1

Mobile

1.2



红外结构光发射器及开孔

通过红外结构光可有效解决白墙等无纹理表面的视觉计算。

摄像头

摄像头传感器镜头，在使用中请注意保护，以避免成像质量下降。

Type-C 接口及固定孔

使用中，插上Type-C数据线后，请使用接口端的螺丝紧固接口，以避免使用中损坏接口，也保证数据连接的稳定性。

手机模组专用 Type-C 线

自带紧固结构

1.3

	Stark
	PCB dimension:58.1x29.5mm
Total	dimension:75.5x34.5x12.9mm
	Up to 30fps
	2560*720;1280*480
	1280*720; 640*480
	3.0*3.0μm
	40.0mm
	Replacable Standard M7
	D:66° H:59° V:35°
	3.3mm
IR	YES
IR	3m
	Color
	0.2-2.8m
	Rolling Shutter
	1.64W@5V DC from USB
	YUYV/MJPG
	Type-C/Micro
	44g
UVC MODE	YES

1.4

*1 Type-c Type-c *1 Type-c Micro usb *1 M1.6X5*4 mini *1 *1

CHAPTER 2

MYNT® EYE 1200 SDK

2.1

Android 5.x ~ Android 8.x

2.2 SDK

zip apk, aar, demo, doc

- https://pan.baidu.com/s/1aMCPwtUkQPZ7I5nemSUAsA
- Google Drive: https://drive.google.com/open?id=1wVp4xqqgjidPQyzzW1Tmibbw4yY5p4sv

2.3 SDK

1. SDK aar libs
2. build.gradle aar

```
dependencies {
    implementation fileTree(include: ['*.aar'], dir: 'libs')
    ....
}
```

2.4 SDK

2.4.1 v1.2.5 - 2019-07-01

-
-
- app ,
-

2.4.2 v1.2.4 - 2019-05-27

- sample
-
- close crash
- crash

2.4.3 v1.2.3 - 2019-04-10

-
-
- &
- FramData

2.4.4 v1.2.2 - 2019-03-28

- ANR
-
- SDK

2.4.5 v1.2.1 - 2019-02-26

- D-1000 IMU

2.4.6 v1.2.0 - 2019-02-26

- D-1200
- D-1000 IMU | IR

CHAPTER 3

MYNT® EYE 1200 SDK

3.1 SDK

3.1.1 SDK

```
MYNTCamera.getSDKVersion();
```

3.1.2 SDK

```
MYNTCamera.getSDKBuildTime();
```

3.2

3.2.1 USBMonitor

```
mUSBMonitor = USBMonitor(mContext, object : USBMonitor.IUSBMonitorListener {

    override fun didFoundCamera(camera: MYNTCamera) {
        //
    }

    override fun didDettach(camera: MYNTCamera) {
        //
    }

    override fun didConnectedCamera(camera: MYNTCamera) {
```

(continues on next page)

(continued from previous page)

```
        //
    }

    override fun didDisconnectedCamera(camera: MYNTCamera) {
        //
    }

})
```

3.2.2 USBMonitor USB

```
mUSBMonitor?.register()
```

3.2.3 USBMonitor USB

```
mUSBMonitor?.unregister()
```

3.2.4 USBMonitor

```
mUSBMonitor?.destroy()
```

3.3

3.3.1 Camera

```
mCamera?.setCameraListener(object : MYNTCamera.ICameraListener {

    override fun didConnectedCamera(camera: MYNTCamera?) {

    }

    override fun didDisconnectedCamera(camera: MYNTCamera?) {

    }
})
```

3.3.2

```
mCamera?.connect()
```

3.3.3

```
//
mCamera?.open()
//  IR
mCamera?.irCurrentValue = IR_DEFAULT_VALUE

backgroundHandler?.post {
    if (mCamera == null) return@post
    //
    mColorSurface = Surface(colorTextureView.surfaceTexture)
    //
    mDepthSurface = Surface(depthTextureView.surfaceTexture)

    mCamera?.setPreviewDisplay(mDepthSurface, MYNTCamera.Frame.Depth)
    mCamera?.setPreviewDisplay(mColorSurface, MYNTCamera.Frame.Color)
    //     (480 / 720)
    mCamera?.setPreviewSize(previewSize.height)
    //     8bit / 11bit
    mCamera?.setDepthType(depthType)
    //
    mCamera?.setFrameCallback { data ->
        if (data.flag == FrameData.DEPTH) {
            //
        }
        if (data.flag == FrameData.COLOR) {
            //
        }
    }
    //
    mCamera?.start(MYNTCamera.Source.ALL)
}
```

3.4

3.4.1

```
mCamera?.setFrameCallback { data ->
    if (data.flag == FrameData.DEPTH) {
        //
    }
    if (data.flag == FrameData.COLOR) {
        //
    }
}
```

3.5 IMU

3.5.1 IMU IMU

```
mCamera?.setImuCallback { data ->

    if (data.flag == ImuData.ACCELEROMETER) {
        runOnUiThread {
            accTextView.text = String.format("acc: x -> %.2f, y -> %.2f, z -> %.2f,␣
↪timestamp -> %d, temperature -> %.2f", data.value[0], data.value[1], data.value[2],␣
↪data.timestamp, data.temperature)
        }
    }
    if (data.flag == ImuData.GYROSCOPE) {
        runOnUiThread {
            gyroTextView.text = String.format("gyro: x -> %.2f, y -> %.2f, z -> %.2f,␣
↪timestamp -> %d, temperature -> %.2f", data.value[0], data.value[1], data.value[2],␣
↪data.timestamp, data.temperature)
        }
    }
}
```

3.6

3.6.1

```
val rectifyLogData = mCamera?.rectifyLogData
```

CHAPTER 4

API

4.1 USBMonitor

4.1.1

```
public void register()
```

4.1.2

```
public void unregister()
```

4.1.3

```
public void destroy()
```

4.2 MYNTCamera

4.2.1

```
public boolean isConnected()
```

4.2.2

```
public boolean isStart()
```

4.2.3

```
public boolean isOpen()
```

4.2.4 USB3.0

```
public boolean getIsUSB3()
```

4.2.5 IMU

```
public boolean isIMUSupported()
```

4.2.6

```
public String getSerialNumber()
```

4.2.7

```
public String getName()
```

4.2.8

```
public int getCameraType()
```

4.2.9

```
public void setCameraListener(ICameraListener callback)
```

4.2.10 IMU IMU

```
public void setImuCallback(IIMUCallback callback)
```

4.2.11

```
public void setFrameCallback(IFrameCallback callback)
```

4.2.12

```
public void connect()
```

4.2.13

```
public int open()
```

4.2.14

```
public void close()
```

4.2.15 IMU / VIDEO / ALL

```
public boolean start(Source source)
```

4.2.16

```
public void destroy()
```

4.2.17 8bit / 11bit

```
public void setDepthType(short depthType)
```

4.2.18

```
public short getDepthType()
```

4.2.19 480 / 720

[D-1000 720]

```
public void setPreviewSize(int height)
```

4.2.20

```
public int getPreviewWidth()
```

4.2.21

```
public int getPreviewHeight()
```

4.2.22 Surface

```
public void setPreviewDisplay(Surface surface, Frame frame)
```

4.2.23 UVC FPS

```
public double getUVCFPS(Frame frame)
```

4.2.24 Preview FPS

```
public double getPreviewFPS(Frame frame)
```

4.2.25

```
public RectifyLogData getRectifyLogData()
```

4.2.26 IR

```
public boolean isIRSupported()
```

4.2.27 IR

```
public int setIRCurrentValue(int value)
```

4.2.28

```
public FrameData getColorFrameData()
```

4.2.29

```
public FrameData getDepthFrameData()
```

4.2.30 IR

```
public int getIRCurrentValue()
```

4.2.31 IR

```
public int getIRMinValue()
```

4.2.32 IR

```
public int getIRMaxValue()
```

4.2.33 mm

```
public int getDistanceValue(int index)
```

4.2.34

```
public boolean getAEStatusEnabled()
```

4.2.35

```
public void setEnableAE()
```

4.2.36

```
public void setDisableAE()
```

4.2.37

```
public boolean getAWBStatusEnabled()
```

4.2.38

```
public void setEnableAWB()
```

4.2.39

```
public void setDisableAWB()
```

4.2.40

```
public void setEnableFrameFPS(boolean enable, int camera_switch)
```

4.2.41

```
public void savePointCloud(final FrameData colorFrameData,
                           final FrameData depthFrameData,
                           final String filePath,
                           Boolean hasColor)
```

4.3 ImuData IMU

4.3.1

```
/**
 * Data type
 *
 * 1: accelerometer
 * 2: gyroscope
 *
 * */
public int flag;
```

4.3.2

```
public long timestamp;
```

4.3.3

```
public double temperature;
```

4.3.4

```
/**
 * Imu accelerometer data for 3-axis: X, Y, X.
 * Imu gyroscope data for 3-axis: X, Y, Z.
 *
 * */
public double value[];
```

4.4 FrameData

4.4.1

```
/**
 *
 * FrameData.COLOR
 * FrameData.DEPTH
 *
 * */
public int flag;
```

4.4.2

```
public int width;
```

4.4.3

```
public int height;
```

4.4.4

```
/**
 * MYNTCamera.DEPTH_DATA_11_BITS
 * MYNTCamera.DEPTH_DATA_8_BITS
 *
 * */
public int depthType;
```

4.4.5 bitmap

```
public Bitmap convert2Bitmap()
```

4.4.6 flag DEPTH

```
public int getDistanceValue(int index)
```

4.4.7 flag DEPTH

```
public int getDistanceValue(int x, int y)
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search