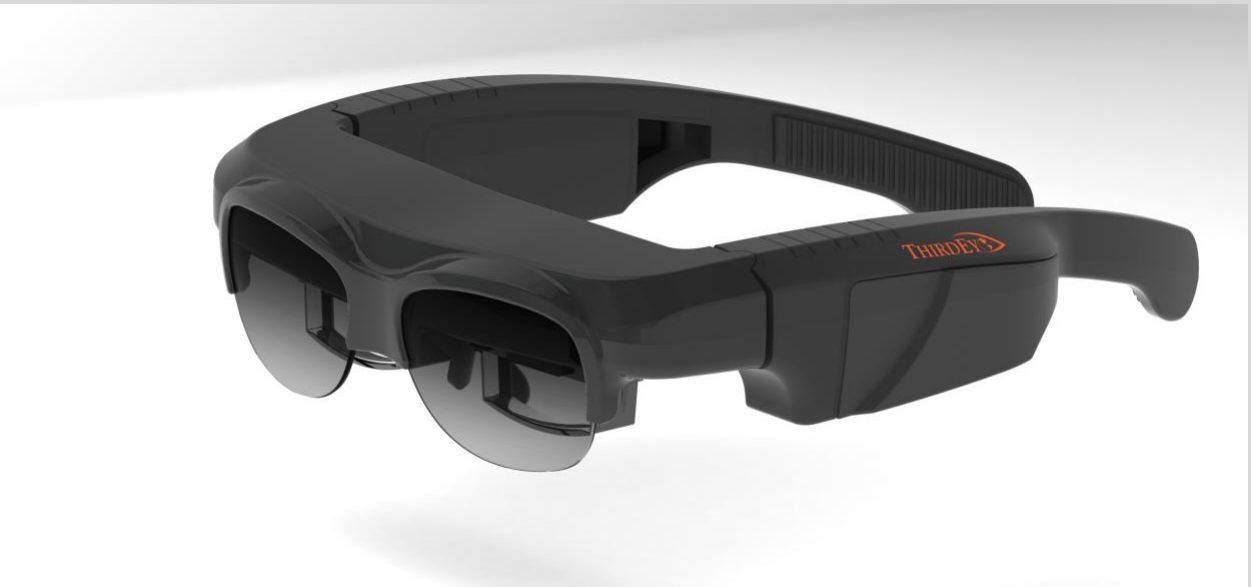
THIRDEYE

X1 Smart Glasses Spec Sheet

Ruggedized Military Technology for the Commercial World

X1 Augmented Reality SmartGlasses™ Developer Guide



Index

1. ThirdEyeOS Software App Development

1.1 Android Platform: The X1 runs on Android 7.0.1 API level 24.

1.2 Development Platforms: Apps Built in Android Studio, Unity or other AR development platforms are compatible with the X1 if they are built on Android 5.0+. We recommend Unity3D if you are using powerful graphics (3D) in your application. Wikitude, Kudan and Vuforia are also compatible if they are built to our Android platform and are recommended for AR apps that require (location markers, SLAM).

Download Android Studio: https://developer.android.com/studio/index.html

Download Unity3D: https://unity3d.com/get-unity/download

Download Wikitude: https://www.wikitude.com/download/

1.3 GPU: The X1 uses an Adreno 530 GPU which supports OpenGL ES 3.1 AEP, OpenCL 2.0, Direct3D 11.1 (FL 11_1)

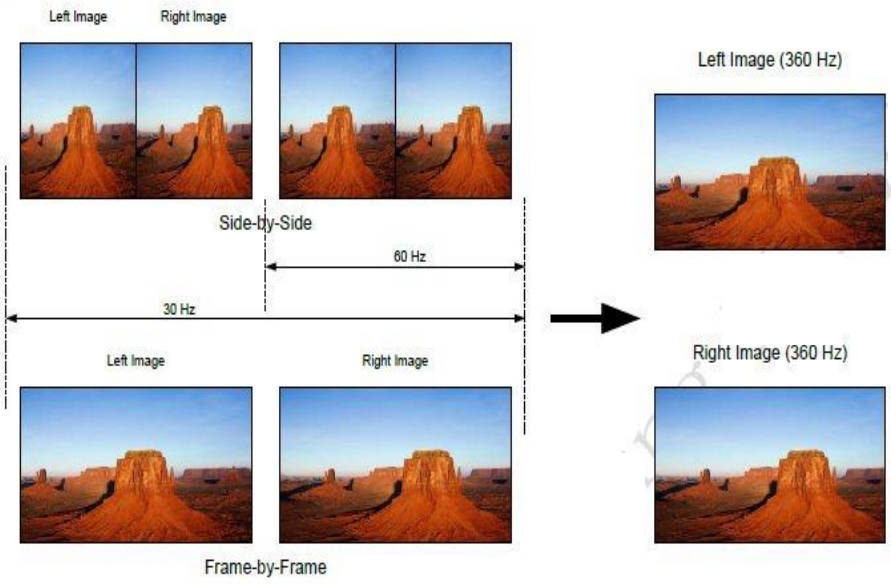
1.4 Connecting to Computer: The Computer will recognize the X1 as an Android Device. (If you are using a Mac then download/open Android File Transfer https://www.android.com/filetransfer/to recognize the device)

1.5 3D Display: The X1 supports 3D Stereoscopic Renderings.

Frame-by-Frame and Side-by-Side formats are both available.

3D FORMAT CONVERSION

It supports to converter from 2D image to 3D image. Frame-by-Frame and Side-by-Side format are available. When Pin No. D11 is toggled, Display format is converted from 2D image to 3D image. In case of Frame-by-Frame mode, Pin No. P15 indicates whether current frame is for left or right.



1.6 Display: The X1 uses a 1280x720 resolution Display screen.
If you are developing on android phone, make sure to always use landscape mode. Make Background Screens Black and overlay information on top. The X1 is approximately a virtual 90'' screen focused at 10 feet.

1.7 Sensors: All sensors in the standard Android API can be used.

- If you are developing just using an Android phone — switch the android phone into landscape mode (horizontal) and place it to your face (similar to VR mode). There are some differences in IMU orientation between an Android Phone and X1 (see below) but this will enable you to move around your head and simulate AR movement.

SENSOR_MANAGER

Rotation Vector Data

The sensor data values returned for the rotation vector **are reversed in direction** from the Android standard. The axis values represent rotations toward these directions:

	X1 Smart-Glass	Android
X-axis	points left	points right
Y-axis	points front of screen (north)	points Up
Z-axis	points toward ground	points outside to front screen

For both X1 and standard Android, the Z-axis value is perpendicular to the ground plane.

For the below picture: rotate 180 degrees for our IMU-> that is what is listed in above ^ table.

15 ORIENTATION OF AXES

Figure 12 and Figure 13 show the orientation of the axes of sensitivity and the polarity of rotation. Note the pin 1 identifier (•) in the figures.

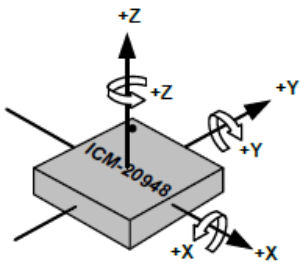


Figure 12. Orientation of Axes of Sensitivity and Polarity of Rotation

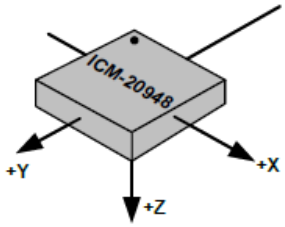


Figure 13. Orientation of Axes of Sensitivity for Magnetometer

See the Android https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-identifyfor other best practices when using Android sensors.

1.8 Bluetooth: The X1 supports Bluetooth so a wireless keyboard/touchpad that is compatible with Android can be used. You can also connect to a phone's application via Bluetooth and run it on the X1. However, this is not recommended as we prefer the user to be hands-free.

1.9 UI Controls (No Touchpad on X1): The X1 is designed for a pure AR experience —meaning that we do not want users to have to use a touchpad while controlling the

cursor (using your hands on a smartglass touchpad is an unnatural movement — unlike scrolling on a phone, which is a natural movement).
The X1 comes with the following buttons: “Enter”; “Back/Home”; “Volume+”; “Volume-”; “Mute”; and “Power” — each of which can be accessed through standard Android calls.

When developing User Interface (UI) Controls, we recommend either of the following approaches:

1.9.1 Our ThirdEye OS uses head motion as the method to move the cursor. This is written as a service in our base controls.

To click a button, make the user press the “Enter” button. To scroll down, utilize rapid head movement up or down.

1.9.2 You also have the option of overriding our cursor and using your own custom head motion cursor service. If you want a pure hands-free User experience you could do the following: to click, make the user stare at the button with the cursor on top for 2-3 seconds and have a progress bar on top. This would enable a pure hands-free UI. Therefore, the base controls of the X1 are **via the user’s head motions and virtual finger-motion**.

1.11 User Interface

See 1.10 for UI controls in developing app.

1.12 Converting existing Android app for X1

If you have already developed an Android app and want to submit it to the ThirdEye App store see below.

1.12.1 **The X1 is NOT a Google Mobile Services certified device so as a developer, your apps cannot use services that require Google MS authentication (like Google Maps or Google Play).**

1.12.2 The Background screens should be Black so that the app is visible under different lighting conditions. It is highly recommended to use a **bright orange** rgb(255,165,0) as the foreground color and to use **black** as the background color. This color contrast is ideal for Augmented Reality usage in different lighting conditions.

1.12.3 Make sure the screen size is adjusted from your phone to the X1. Also re-check the sensors orientation and adjust app as needed.

1.12.4 Try not to keep same screen for a long time (or utilize a sleep function).

1.12.5 Make sure the text is recognizable under different lighting conditions. Use a larger text size and color combination. (this is most easily achieved through point 2.2— use orange text.

1.12.6 Make sure controls can be controlled via head-motion and are not touchpad dependent.

1.12.7 Make sure IMU orientation is oriented per 1.7.

Make sure all sensors being used are available on X1— see
https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-identify

1.13.Voice/Text Input

The X1 has a microphone to capture voice/sound.
When designing an application to use the ThirdEye voice input function, keep in mind that voice input may be hard to interpret correctly when there is ambient noise or wind. We recommend using sound stabilization to block excess inputs or that you suggest an alternative input method to users when outdoors or in public places.

---How to VoiceInput for Developers---

All Android.media audio actions can be called by developers except for Google Play specific functions.

---1.14 Check out https://developer.android.com/studio/run/emulator.html#avds to run as an emulator
https://developer.android.com/studio/run/managing-avds.html create your AVD----

1.15 Brightness To control the X1 screen brightness, you can use the standard android sensor functions for brightness — specified in
https://developer.android.com/reference/android/provider/Settings.System.html

1.16 Camera- **Make sure to call the camera 2 api if making calls to camera**. The X1 uses https://developer.android.com/reference/android/hardware/camera2/package-summary.html

1.17. Submit apps to app store- Developers can make profile at developers.thirdeyegen.com and submit their apps to App Store.

Make sure your package ID , version number & version code correlates to what is in your .apk.

2. Development Overview/Restricions-

Refer to 1.10-1.15 notes.

3. Android Debug Bridge (ADB) for ThirdEyeOS

3.1 Android Debug Bridge (ADB) is a useful tool to communicate between Android devices and Windows/Mac computers. ADB is a command-line utility included with Google's Android SDK. ADB can control your device over USB from a computer, copy files back and forth, install and uninstall apps, run shell commands, and more. ADB can be used to identify apps or services that may be consuming too much battery power and other use cases.

3.2 Make sure to first set up the Android SDK if you have not already. Head to the Android SDK download page at https://developer.android.com/studio/index.html and scroll down to "SDK Tools Only", which is a set of tools that includes ADB. Download the ZIP file for your platform and unzip it wherever you want to store the ADB files — they're portable, so you can put them anywhere you want.

Start the SDK Manager EXE and deselect everything except "Android SDK Platform-tools". Click the Install button. This downloads and installs the platform-tools package, which contains ADB and other utilities.

3.3 For more info on USB debugging in Android Studio check out https://developer.android.com/studio/run/device.html#setting-up

3.4 Enable USB Debugging on Smart glass
To enable USB Debugging on Android 5.0+ is the same as Android 4.2.x.
Settings > About Phone > Build number > Tap it 7 times to become developer; click on "enable USB debugging"

Settings > Developer Options > USB Debugging.

3.5 Setting up ADB on Windows

1. Turn on X1
2. Connect the X1 to your computer via USB.
3. Navigate to ThirdEyeGen/Developers directory and copy the windows_adb folder to the folder on the computer where you want the drivers to be installed (choose folder for where to install).
4. Open the Device Manager and locate the X1 smartglass. Right click on it and select Update Driver Software
5. Choose "Browse my computer" for driver software.
6. Browse to the path usb_driver folder located in the windows_adb folder.
7. Ensure "Include subfolders" is checked.
8. Press Next.

9. If Windows warns of an unverified publisher, choose “Install this driver software anyway”.
10. Open up command and navigate to the usb-driver folder.
11. Run the command adb.exe X1 to test if ADB recognizes the device.
12. ADB should detect the X1 and display its serial number.

3.6 Setting up ADB on Mac

1. Turn on X1.
2. Download and Install Android File Transfer per step 1.4.
3. Connect X1 and open Android File Transfer.
4. Browse to ThirdEyeGen/Developers and copy the mac_adb folder to the desired location on computer.
5. Launch terminal and navigate to the mac_adb directory.
6. Run the following command adb devices.
7. ADB should detect the X1 and display its serial number

4. Initial Screens

The ThirdEyeOS comes with the following — the base Android apps as well as the ThirdEye App Store.

-For certain enterprise solutions, additional custom apps also appear upon startup or the X1 loads directly into an app. But for most use cases, the above are the base apps that appear upon every startup.

5. Design Guidelines

- If you are developing just using an Android phone — switch the android phone into landscape mode (horizontal) and place it to your face (similar to VR mode). There are some differences in IMU between an Android Phone and X1 (see below) but this will enable you to move around your head and simulate AR movement.

Background – Make the background screen black – this will make it easier for user to view information like on phone. Without a black background screen, in high brightness it may be tough for users to view text. The best color combination is an orange foreground on a black background.

Head Movement — Keep user fatigue at a minimum. That means when rotating your head, keep the screen images/text stable and do not have them rapidly change, which would make the user dizzy.

Multiple Screens and Increased Development Space — As a developer, utilize the extra screen availability for your app. i.e. the screen size is not limited like for a phone. If users rotate their heads while wearing X1, the users can see different screens, data, etc. You can place data anywhere users will look —- that is the big difference between X1 smartglass and standard android phone development.

API use — Utilize the provided ThirdEye libraries in the way they are documented. This ensures that your app utilizes battery life and the app functions smoothly. If it is natural to nod or select in your app, then utilize those API calls as ThirdEye provides them in the SDK. As a developer, you can save a lot of development time and code your app more efficiently by properly using ThirdEye's provided libraries.

Text Size — Make the text size as large and clear as possible. We recommend 30pixels for Font size. Utilize our ScrollingAPI to scroll. We do not recommend long pieces of continuous text.

Scrolling —Use the ScrollingAPI in the ThirdEye SDK.

When the user is moving around, you should use larger fonts that are easier to read. You should also display less text overall to avoid distraction and interfering with the users' view of their surroundings.
Icon Size — The minimum recommended icon size is 20×20 pixels, so that the icon can be easily recognized by most users. For users who are walking or moving around, we recommend an icon size of at least 60×60 pixels. You can track this via SensorManager. Choose icons that are readily recognizable, so that most users will immediately know what they mean (use standard approved symbols). Be aware that outside light can make icons harder to recognize. Applications intended for outdoor use should make use of brighter and larger icons.

Brightness — If certain portions of your app require the user to see the screen text/symbols clearly, we recommend you suggest to users about increasing the brightness (which users would do from “settings”) or to automatically increase brightness in your app by calling that function.

6. Getting Started with Installation

Install the ThirdEye SDK

To start developing apps for X1, you’ll need to install the ThirdEye SDK. The following instruction requires you to have the Android SDK and Android Studio installed on your computer (you may also be using other developer tools).
There are two ways to install the ThirdEye SDK:

- Install through Android SDK Manager
- Install manually

a. Install through Android SDK Manager-

b. When you have finished installing the ThirdEye SDK, you can set up your IDE to begin developing your ThirdEye app.

c. Open IDE and start the Android SDK manager
d. In the Android SDK manager choose “Tools” -> “Manage Add-On Sites”

e. Select the **User Defined Sites** tab and click **New**.
f. Enter this URL in the **Add Add-on Site URL** dialog. https://www.thirdeyegen.com/developers/ThirdEy SDK.xml
g. Click OK. The Android SDK Manager downloads the SmartEyeglass SDK from the Sony server.
h. Click Close and confirm that you want to sort the listed SDKs by API level.
i. Expand the display of available SDKs available under Android 7.0.1 (API 24), select the checkbox for the ThirdEye SDK, then click Install 1 package. This initiates the installation of the ThirdEye SDK.
j.

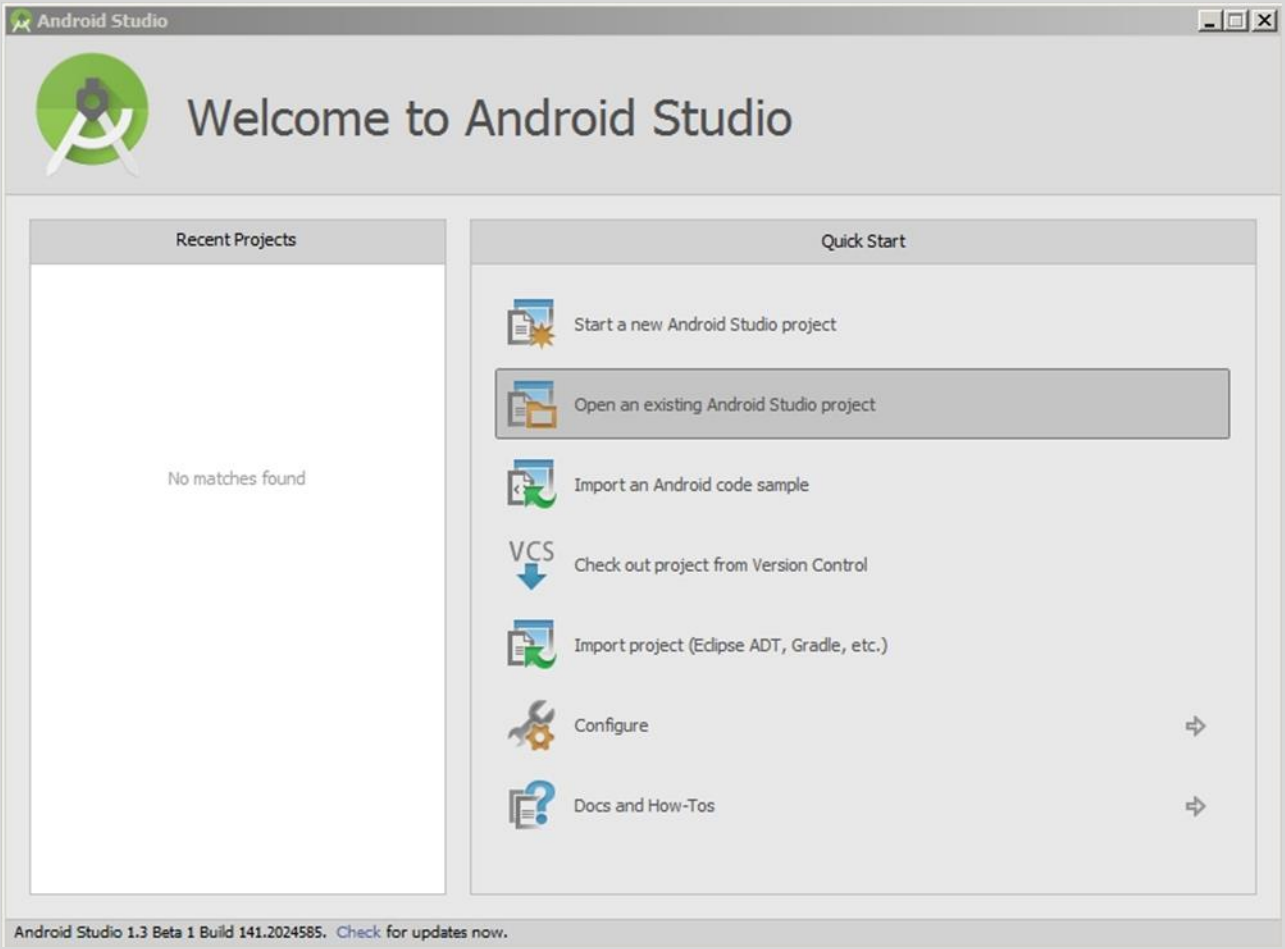
Install ThirdEye SDK Manually —

Install manually
If you're not using the Android SDK Manager, you can still download and install the ThirdEye SDK manually. Simply download the ThirdEye SDK zip file to your computer, and unzip to a location of your choice. The download location you choose is referred to in the documentation as SDK root.
If you choose to install manually, make sure that you have a supported version of the Android API (level 18 or higher) installed.ThirdEye SDK Contents—

The ThirdEye SDK extends the Android SDK with the functionality needed to control and work with the X.
The ThirdEye SDK includes libraries that define the base classes and Android Intents and Content Providers that form the basis for your device-control apps, and also a set of sample apps that demonstrate how to access and control the specific features/APIs of the X1 device. The SDK also includes documentation for the SLAM and other location-tracking AR APIs. To create new applications for the device, you can modify and extend the included samples.
Set-Up Development Environment —

It is very easy to set up your development environment with the X1 project built for Android Studio. You can directly open an existing project.
Please note: Make sure you have Android API 17 onwards installed on your Android SDK Manager.
Set up a workspace in Android Studio

1. From the Quick Start, choose Open an existing Android Studio project.



2. Browse to the SDK_root/samples/ folder (the exact location depends on where you installed the X1 SDK).

(add drawing)

3. Select the directory of 1 sample projects you want to open and click OK

(add drawing)

When import is complete, your workspace should include selected sample project you have opened.

Prepare a test environment —

7. Tutorials

--Coming---