Android SDK

4.1.0.0

Generated by Doxygen 1.8.16

Chapter 1

Seekware Android SDK 4.1

Welcome

The Seek Thermal Android SDK was created for developers who want to use Seek Thermal cameras in their own products. The SDK is designed to be simple to use while also providing access to key capabilities of the camera. We offer the Seek Thermal Android SDK for multiple platforms with a common API.

Supported Cameras

Table 1.1 Figure 1 - Supported Ca

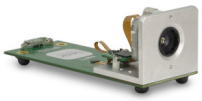
 **Figure 1.1 Compact**	
 **Figure 1.3 Starter Kit**	

Table 1.2 Figure 2 - Supported Platforms

Camera	Image	Speed
Compact PIR-206	206 x 156	< 9Hz
CompactXR PIR-206	206 x 156	< 9Hz
Compact Pro PIR-320	320 x 240	< 9Hz
Compact Pro FF PIR-320	320 x 240	< 18Hz
J2/C2x/SK2x PIR-206	206 x 156	< 18Hz
J3/C3x/SK3x PIR-320	320 x 240	< 18Hz

NOTE: Starter Kits that run at higher frame rates are available on special request.

Seek Thermal SDK API

The class structure of the SDK is modelled after the Android Camera2 API as follows:

Table 1.3 Figure 3 Seekware/Camera2 Class API

Seekware Class	Camera2 Class
SeekCameraManager	CameraManager
SeekCamera	CameraDevice
SeekCameraCaptureSession	CameraCaptureSession
SeekImageView	TextureView/SurfaceView/ImageView

SDK Class Relationship Diagram

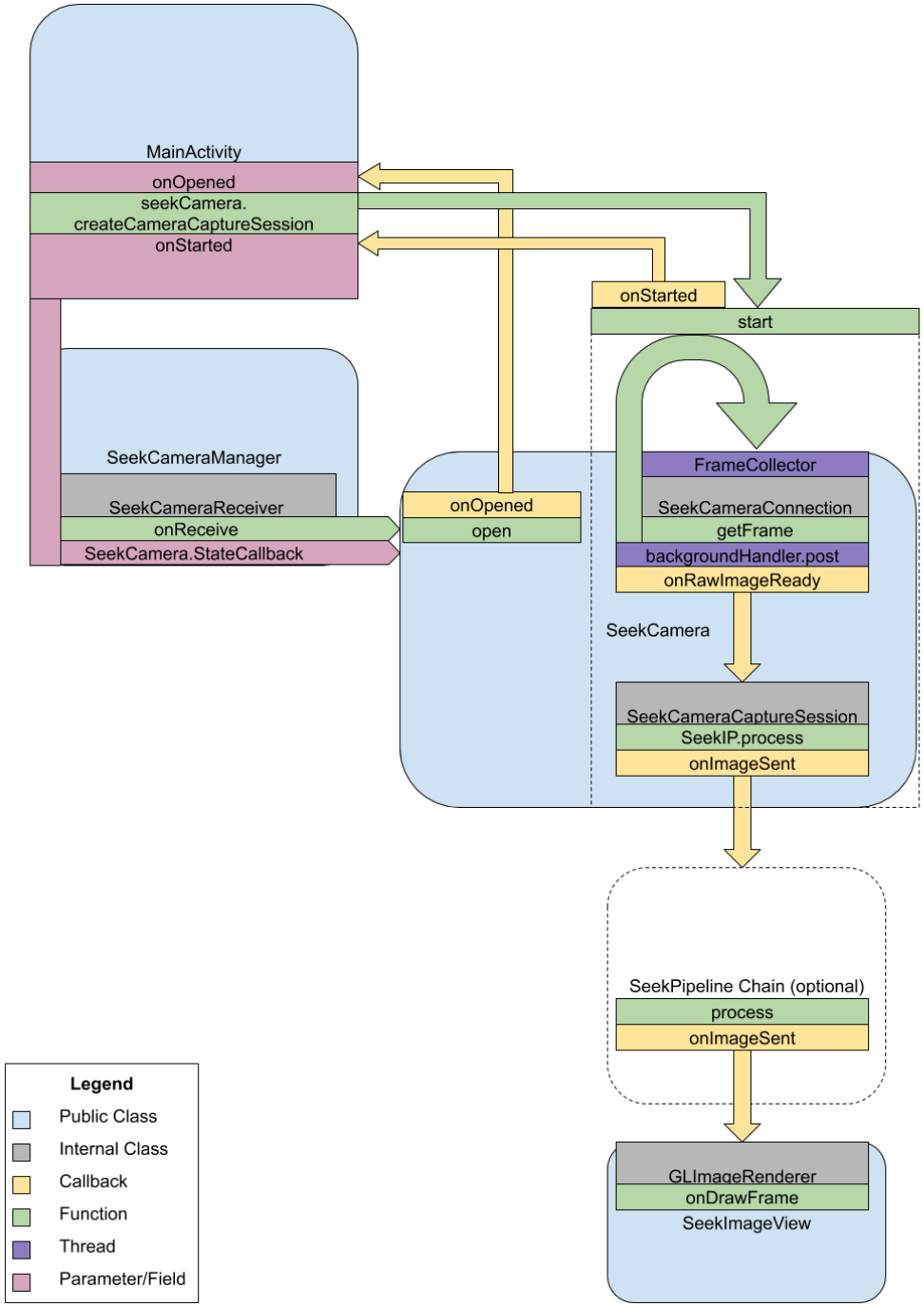


Figure 1.5 Class Relationships

SDK Examples

The Seek Thermal Android Software Development Kit (SDK) contains four examples to help you get started:

1. Seek Hello Example

2. Seek Data Example
3. Seek Upgrade Example
4. Seek Simple Example

SDK 4.1 Release Notes

1. Initial public release
    - Includes cross platform library for USB communication
    - Supports Firmware upgrade functionality
2. Known Bugs/Limitations:
    - No support for Localization strings

Attributions:

- Settings Icon made by Gregor Cresnar from https://www.flaticon.com
- Info Icon from https://en.wikipedia.org/wiki/File:Infobox_info_icon.svg

Build Tools

- Android Studio 3.5.1
- Android 10.0 (Q) API 29
- Android NDK r16b
- Gradle wrapper 5.1.1
- Gradle build tools 3.4.1
- Android SDK Tools 29.0.2
- AndroidX appcompat:appcompat:1.1.0
- AndroidX legacy-preference-v14:1.0.0
- AndroidX constraintlayout:1.1.3
- AndroidX lifecycle-runtime:2.1.0
- AndroidX lifecycle-extensions:2.1.0
- AndroidX vectordrawable:vectordrawable:1.0.1

Chapter 2

Seek Hello Example

This tutorial shows you how to create a new Hello World Android project with Android Studio and the Seek Thermal Android SDK.

1. Create the Android Studio project
2. Change the AndroidManifest.xml file
3. Add Seek Android SDK to project
4. Create the User Interface
5. Write the MainActivity.java code
6. Application Screenshot

Create the Android Studio project

To create your new Android project, follow these steps:

1. In the Welcome to Android Studio window, click Start a new Android Studio project or if you have a project already opened, select File > New > New Project.
2. In the Choose your project window, select Empty Activity and click Next.

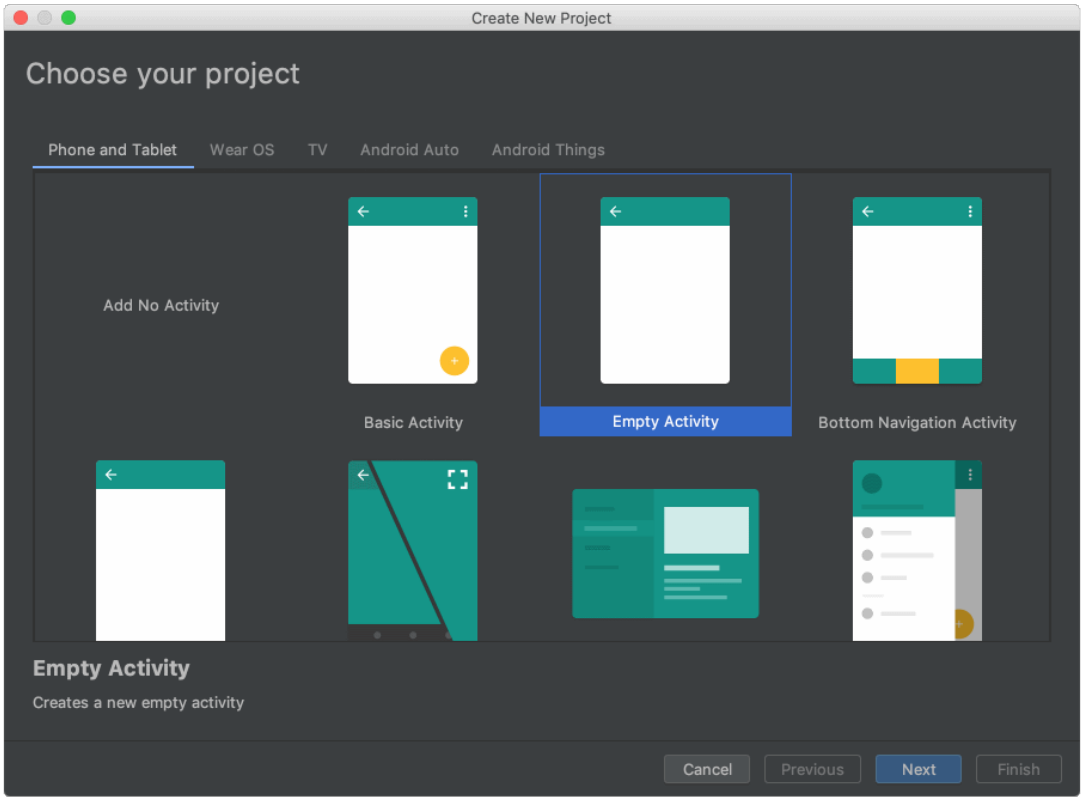


Figure 2.1 New Project

3. In the Configure your project window, complete the following:

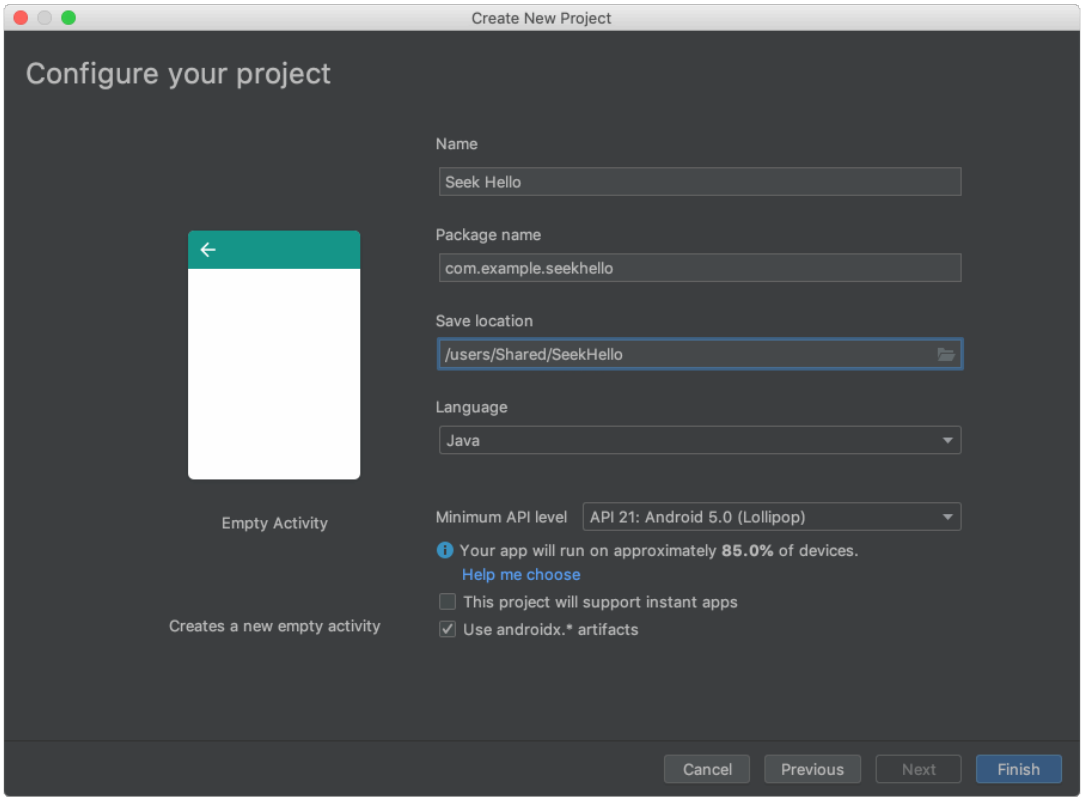


Figure 2.2 Configure Project

- Enter "Seek Hello" in the Name field.
- Enter "com.example.seekhello" in the Package name field.
- If you'd like to place the project in a different folder, change its Save location.
- Select Java from the Language drop-down menu.
- Select the checkbox next to Use androidx.∗ artifacts.
- Click Finish.

Change the AndroidManifest.xml file

Insert the following USB Device Intents to the AndroidManifest.xml file for ".MainActivity":

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
    </intent-filter>
    <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
        android:resource="@xml/seekware_device_filter" />
</activity>
```

Add Seek Android SDK to project

The Seek Thermal Android SDK is delivered as an AAR (Android Archive) for use with Android Studio and Gradle.

1. Open up the project structure by right-clicking on your project and choosing “Open Module Settings” or choosing “File” $>$ “Project Structure...”

2. Click the “+” button in the top left to add a new module.

3. Choose “Import .JAR or .AAR Package” and click the “Next” button.

4. Find your file using the ellipsis button (“. . . ”) beside the “File name” field. Studio will automatically create a subproject name. Click “Finish”.

5. Gradle will sync, which may take a few minutes. Add the new module as a dependency to your app. In the “Project Structure...” window, a new module has appeared representing the SDK. Keep the app’s module selected and click on the Dependencies pane.

6. Use the “+” button at the bottom of the dependencies screen, and choose “module dependency”. The dependencies screen is quite tall, so this button might be invisible at first.

7. The screen that pops up should show the Seekware_Android_4.0. Click “OK”.

Create the User Interface for the Application:

Overwrite the res/layout/activity_main.xml file:

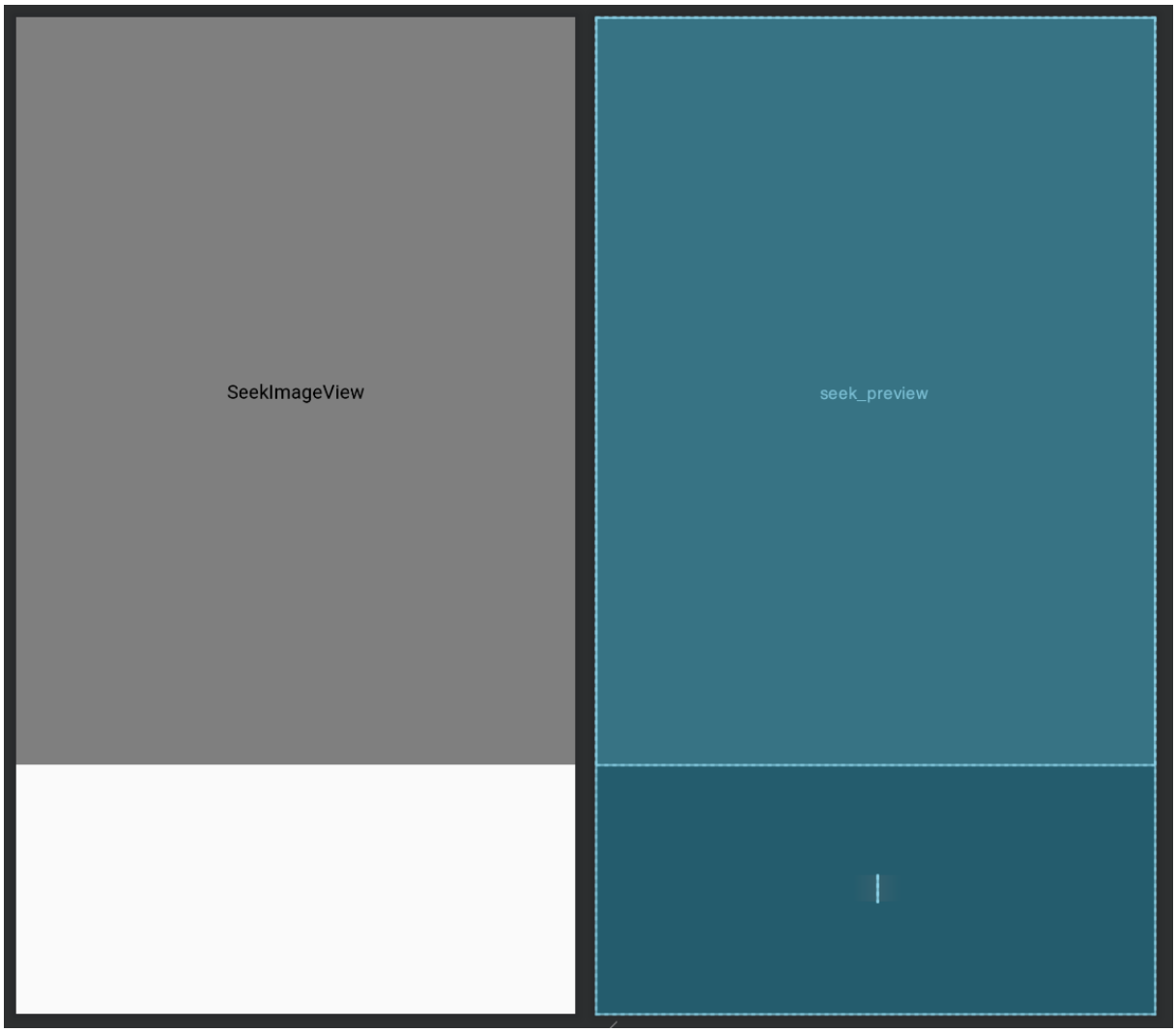


Figure 2.3 User Interface

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
       xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <com.thermal.seekware.SeekImageView
        android:id="@+id/seek_preview"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintDimensionRatio="H, 3:4"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@id/seek_preview"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent">
        <TextView
            android:id="@+id/camera_info"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
            android:textColor="#000000"
            android:gravity="center" />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Write the MainActivity.java code

Add the following handlers to the MainActivity.java source file:

```
import android.content.pm.ActivityInfo;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import com.thermal.seekware.SeekCamera;
import com.thermal.seekware.SeekCameraManager;
import com.thermal.seekware.SeekImageView;
import com.thermal.seekware.SeekLogger;
public class MainActivity extends AppCompatActivity {
    private TextView cameraInfo;
    private SeekCameraManager seekCameraManager;
    private SeekImageView seekImageView;
    private SeekCamera.StateCallback stateCallback = new SeekCamera.StateCallback() {
        @Override
        public void onInitialized(SeekCamera seekCamera) {
        }
        @Override
        public void onOpened(SeekCamera seekCamera) {
            seekCamera.createSeekCameraCaptureSession(seekImageView);
        }
        @Override
        public void onStarted(SeekCamera seekCamera) {
            seekCamera.setColorLut(SeekCamera.ColorLut.TYRIAN);
            cameraInfo.setText(seekCamera.toString());
        }
        @Override
        public void onStopped(SeekCamera seekCamera) {
            cameraInfo.setText("");
        }
        @Override
        public void onClosed(SeekCamera seekCamera) {
        }
        @Override
        public void onMemoryAccess(SeekCamera camera, SeekCamera.MemoryRegion region, int progress){
            SeekLogger.debug("Main Activity", "SeekCamera Memory Read Progress: " + progress);
        }
        @Override
        public void onError(SeekCamera seekCamera, Exception e) {
        }
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        cameraInfo = findViewById(R.id.camera_info);
        seekImageView = findViewById(R.id.seek_preview);
        seekCameraManager = new SeekCameraManager(this, null, stateCallback);
    }
}
```

Application Screenshot

Here is a screenshot of the completed application:

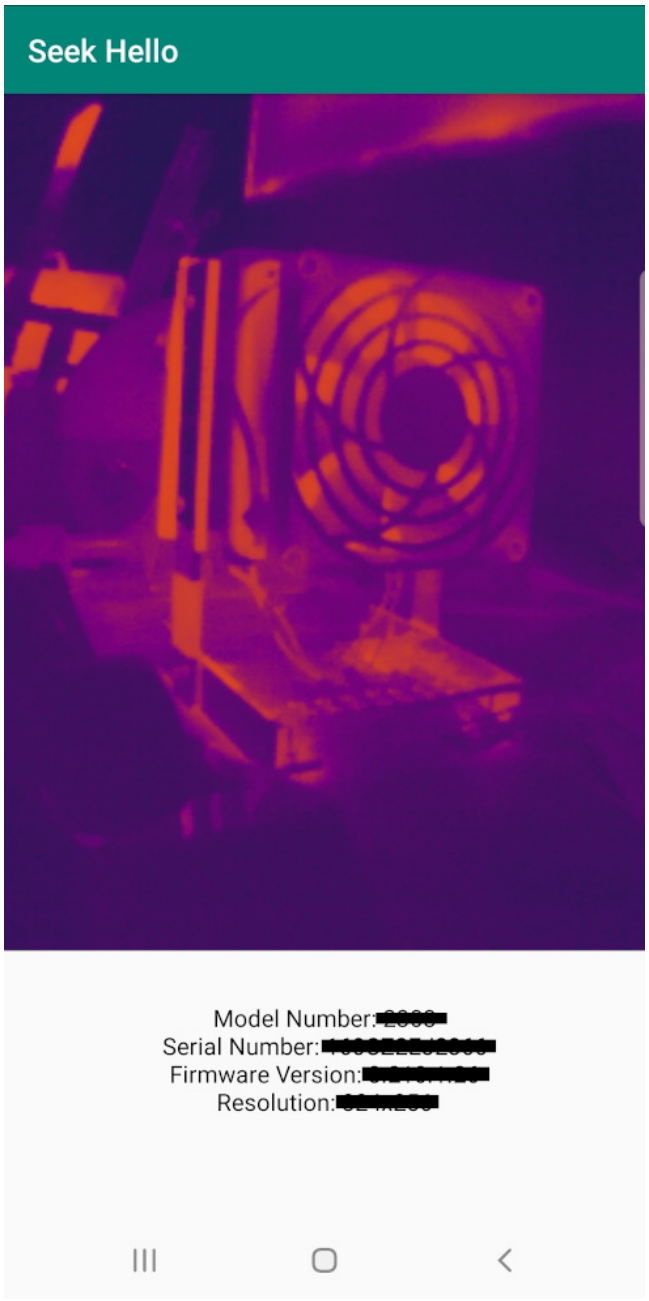


Figure 2.4 Screenshot

Chapter 3

Seek Data Example

This example shows you how to create an application to read data from a Seek Thermal Camera with Android Studio and the Seek Thermal Android SDK.

1. Create the Android Studio project
2. Change the AndroidManifest.xml file
3. Add Seek Android SDK to project
4. Create the User Interface
5. Write the MainActivity.java code
6. Application Screenshot

Create the Android Studio project

To create your new Android project, follow these steps:

1. In the Welcome to Android Studio window, click Start a new Android Studio project or if you have a project already opened, select File $>$ New $>$ New Project.
2. In the Choose your project window, select Empty Activity and click Next.

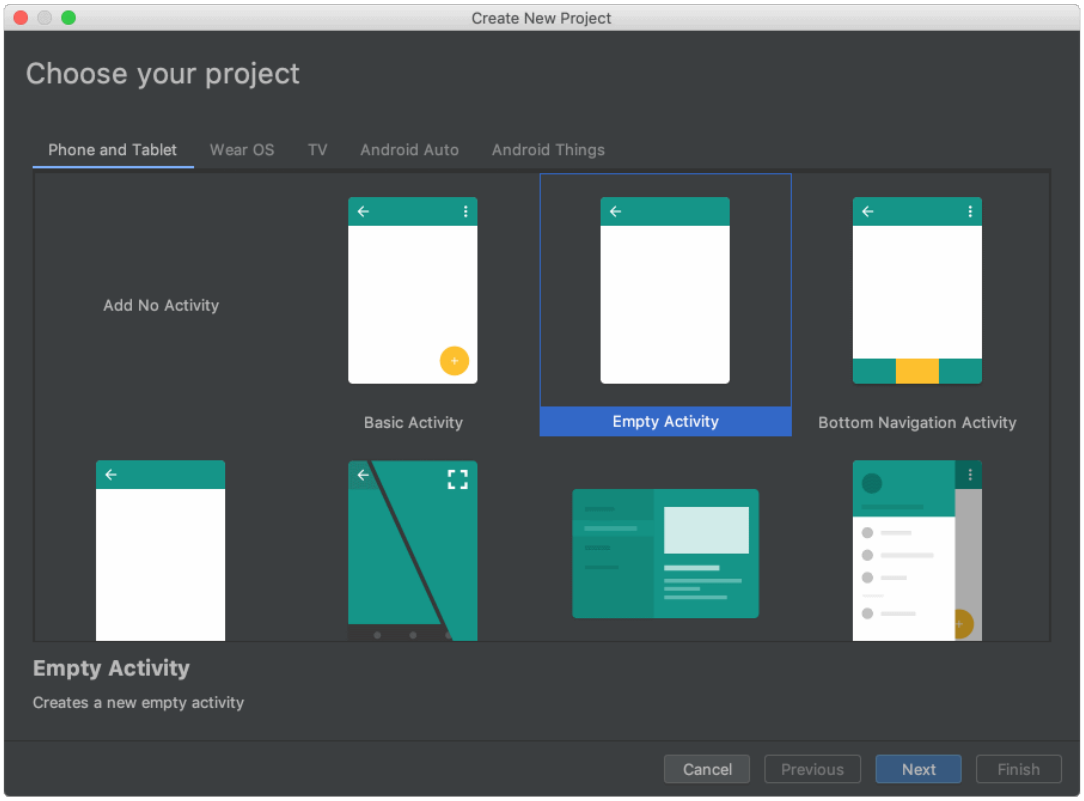


Figure 3.1 New Project

3. In the Configure your project window, complete the following:

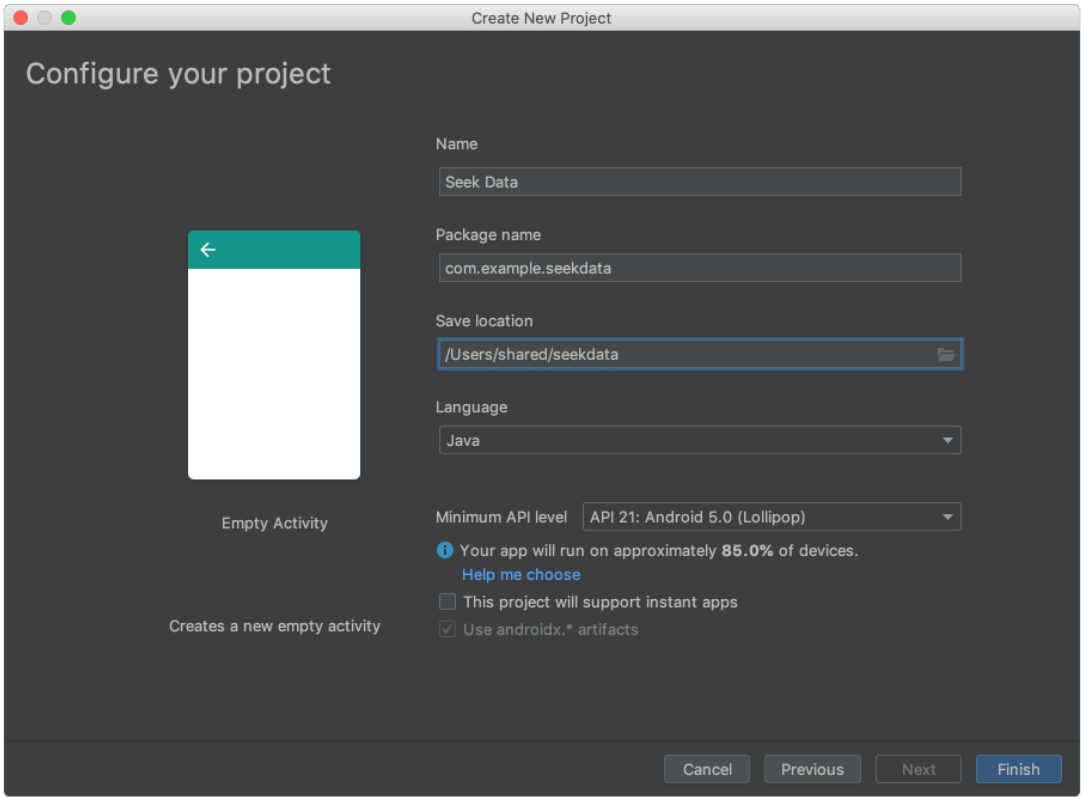


Figure 3.2 Configure Project

- Enter "Seek Data" in the Name field.
- Enter "com.example.seekdata" in the Package name field.
- If you'd like to place the project in a different folder, change its Save location.
- Select Java from the Language drop-down menu.
- Select the checkbox next to Use androidx.∗ artifacts.
- Click Finish.

Change the AndroidManifest.xml file

Insert the following USB Device Intents to the AndroidManifest.xml file for ".MainActivity":

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
    </intent-filter>
    <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
        android:resource="@xml/seekware_device_filter" />
</activity>
```

Add Seek Android SDK to project

The Seek Thermal Android SDK is delivered as an AAR (Android Archive) for use with Android Studio and Gradle.

1. Open up the project structure by right-clicking on your project and choosing “Open Module Settings” or choosing “File” $>$ “Project Structure...”

2. Click the “+” button in the top left to add a new module.

3. Choose “Import .JAR or .AAR Package” and click the “Next” button.

4. Find your file using the ellipsis button (“. . . ”) beside the “File name” field. Studio will automatically create a subproject name. Click “Finish”.

5. Gradle will sync, which may take a few minutes. Add the new module as a dependency to your app. In the “Project Structure...” window, a new module has appeared representing the SDK. Keep the app’s module selected and click on the Dependencies pane.

6. Use the “+” button at the bottom of the dependencies screen, and choose “module dependency”. The dependencies screen is quite tall, so this button might be invisible at first.

7. The screen that pops up should show the Seekware_Android_4.0. Click “OK”.

Create the User Interface for the Application:

Overwrite the res/layout/activity_main.xml file:

Figure 3.3 User Interface

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
       xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/thermography_info"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Write the MainActivity.java code

Add the following handlers to the MainActivity.java source file:

```
package com.example.seekdata;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import com.thermal.seekware.SeekCamera;
import com.thermal.seekware.SeekCameraManager;
import com.thermal.seekware.SeekImage;
import com.thermal.seekware.SeekImageReader;
import com.thermal.seekware.SeekLogger;
public class MainActivity extends AppCompatActivity implements SeekImageReader.OnImageAvailableListener {
    private TextView thermographyInfo;
    private SeekCamera seekCamera;
    private SeekCameraManager seekCameraManager;
    private SeekImageReader seekImageReader;
    private SeekCamera.StateCallback stateCallback = new SeekCamera.StateCallback() {
        @Override
        public void onInitialized(SeekCamera seekCamera) {
        }
        @Override
        public void onOpened(SeekCamera sc) {
            seekCamera = sc;
            seekCamera.createSeekCameraCaptureSession(seekImageReader);
        }
        @Override
        public void onStarted(SeekCamera seekCamera) {
        }
        @Override
        public void onStopped(SeekCamera seekCamera) {
        }
        @Override
        public void onClosed(SeekCamera seekCamera) {
        }
        @Override
        public void onMemoryAccess(SeekCamera camera, SeekCamera.MemoryRegion region, int progress){
            SeekLogger.debug("Main Activity", "SeekCamera Memory Progress: " + progress);
        }
        @Override
        public void onError(SeekCamera seekCamera, Exception e) {
        }
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        thermographyInfo = findViewById(R.id.thermography_info);
        seekImageReader = new SeekImageReader();
        seekImageReader.setOnImageAvailableListener(this);
        seekCameraManager = new SeekCameraManager(this, null, stateCallback);
    }
    @Override
    public void onImageAvailable(final SeekImage seekImage) {
        runOnUiThread(() -> {
            String text = seekCamera.toString() + "\nMin: " +
       seekImage.getThermography().getMinTemp().toString() + "\n" +
                    "Spot: " + seekImage.getThermography().getSpotTemp().toString() + "\n" +
                    "Max: " + seekImage.getThermography().getMaxTemp().toString();
            thermographyInfo.setText(text);
        });
    }
}
```

Application Screenshot

Here is a screenshot of the completed application:

Figure 3.4 Screenshot

Chapter 4

Seek Upgrade Example

This example shows you how to create an application to upgrade the firmware on a Seek Thermal Camera with Android Studio and the Seek Thermal Android SDK.

1. Create the Android Studio project
2. Change the AndroidManifest.xml file
3. Add Seek Android SDK to project
4. Create the User Interface
5. Write the MainActivity.java code
6. Application Screenshot

Create the Android Studio project

To create your new Android project, follow these steps:

1. In the Welcome to Android Studio window, click Start a new Android Studio project or if you have a project already opened, select File > New > New Project.
2. In the Choose your project window, select Empty Activity and click Next.

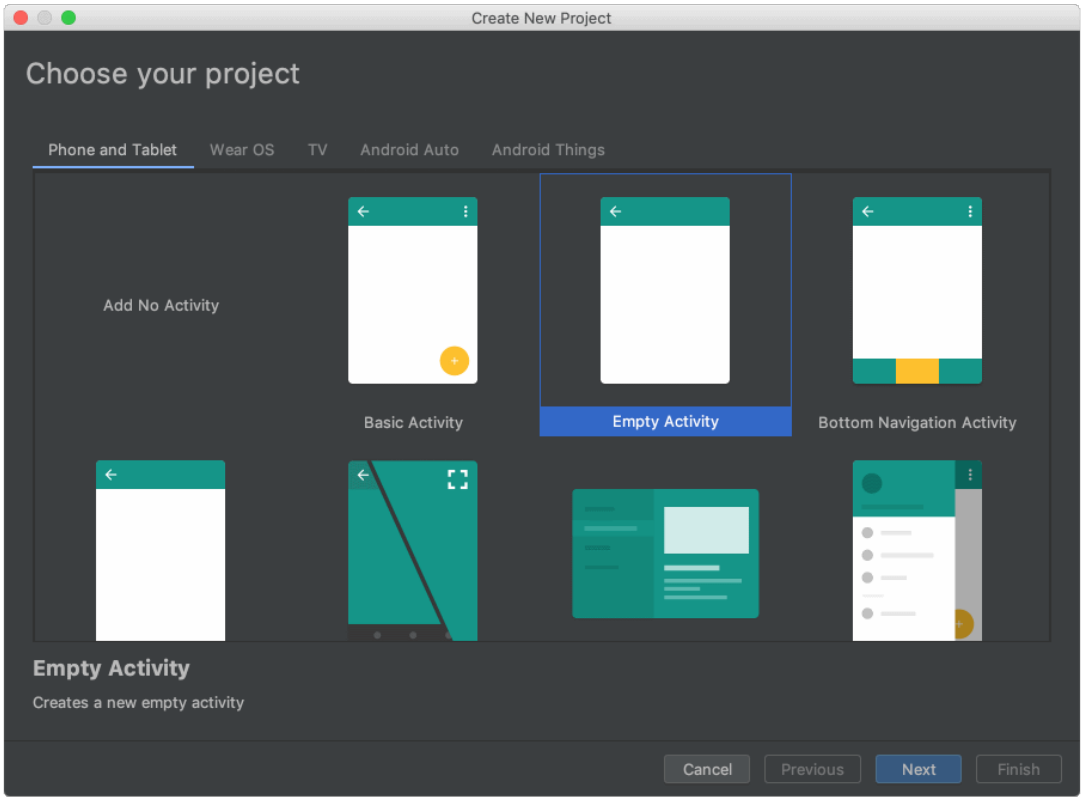


Figure 4.1 New Project

3. In the Configure your project window, complete the following:

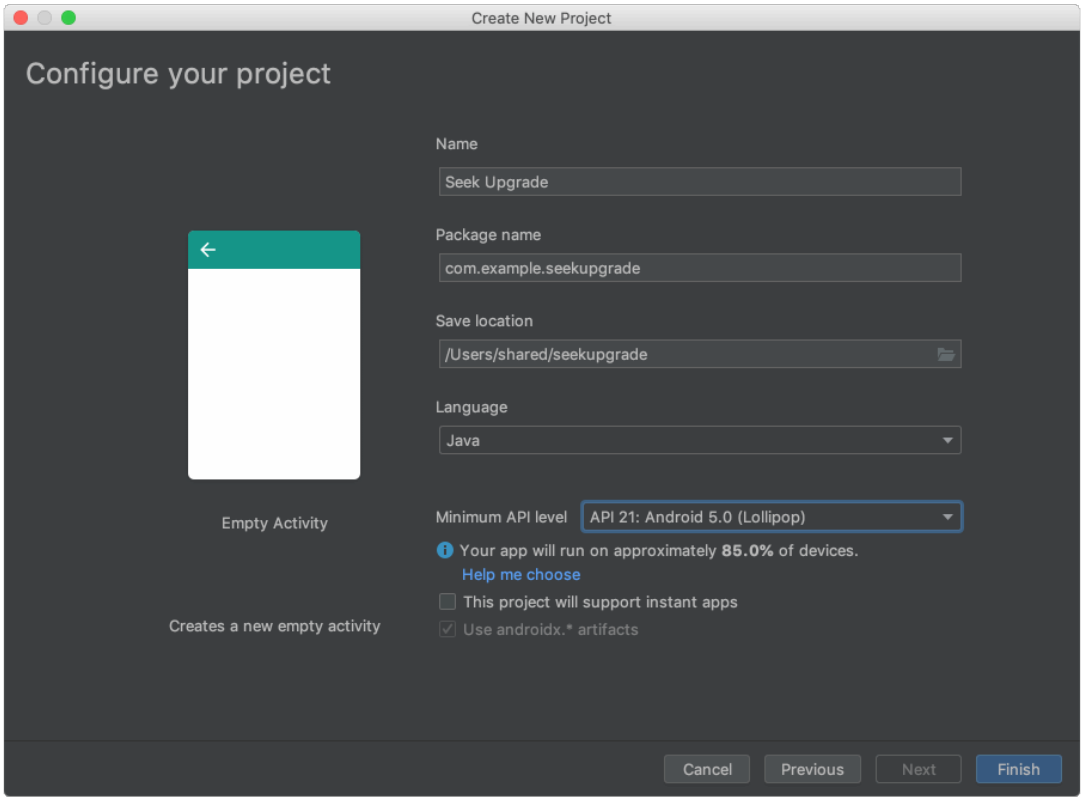


Figure 4.2 Configure Project

- Enter "Seek Upgrade" in the Name field.
- Enter "com.example.seekupgrade" in the Package name field.
- If you'd like to place the project in a different folder, change its Save location.
- Select Java from the Language drop-down menu.
- Select the checkbox next to Use androidx.∗ artifacts.
- Click Finish.

Change the AndroidManifest.xml file

Insert the following USB Device Intents to the AndroidManifest.xml file for ".MainActivity":

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
    </intent-filter>
    <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
        android:resource="@xml/seekware_device_filter" />
</activity>
```

Add Seek Android SDK to project

The Seek Thermal Android SDK is delivered as an AAR (Android Archive) for use with Android Studio and Gradle.

1. Open up the project structure by right-clicking on your project and choosing “Open Module Settings” or choosing “File” $>$ “Project Structure...”

2. Click the “+” button in the top left to add a new module.

3. Choose “Import .JAR or .AAR Package” and click the “Next” button.

4. Find your file using the ellipsis button (“. . . ”) beside the “File name” field. Studio will automatically create a subproject name. Click “Finish”.

5. Gradle will sync, which may take a few minutes. Add the new module as a dependency to your app. In the “Project Structure...” window, a new module has appeared representing the SDK. Keep the app’s module selected and click on the Dependencies pane.

6. Use the “+” button at the bottom of the dependencies screen, and choose “module dependency”. The dependencies screen is quite tall, so this button might be invisible at first.

7. The screen that pops up should show the Seekware_Android_4.0. Click “OK”.

Create the User Interface for the Application:

Overwrite the res/layout/activity_main.xml file:

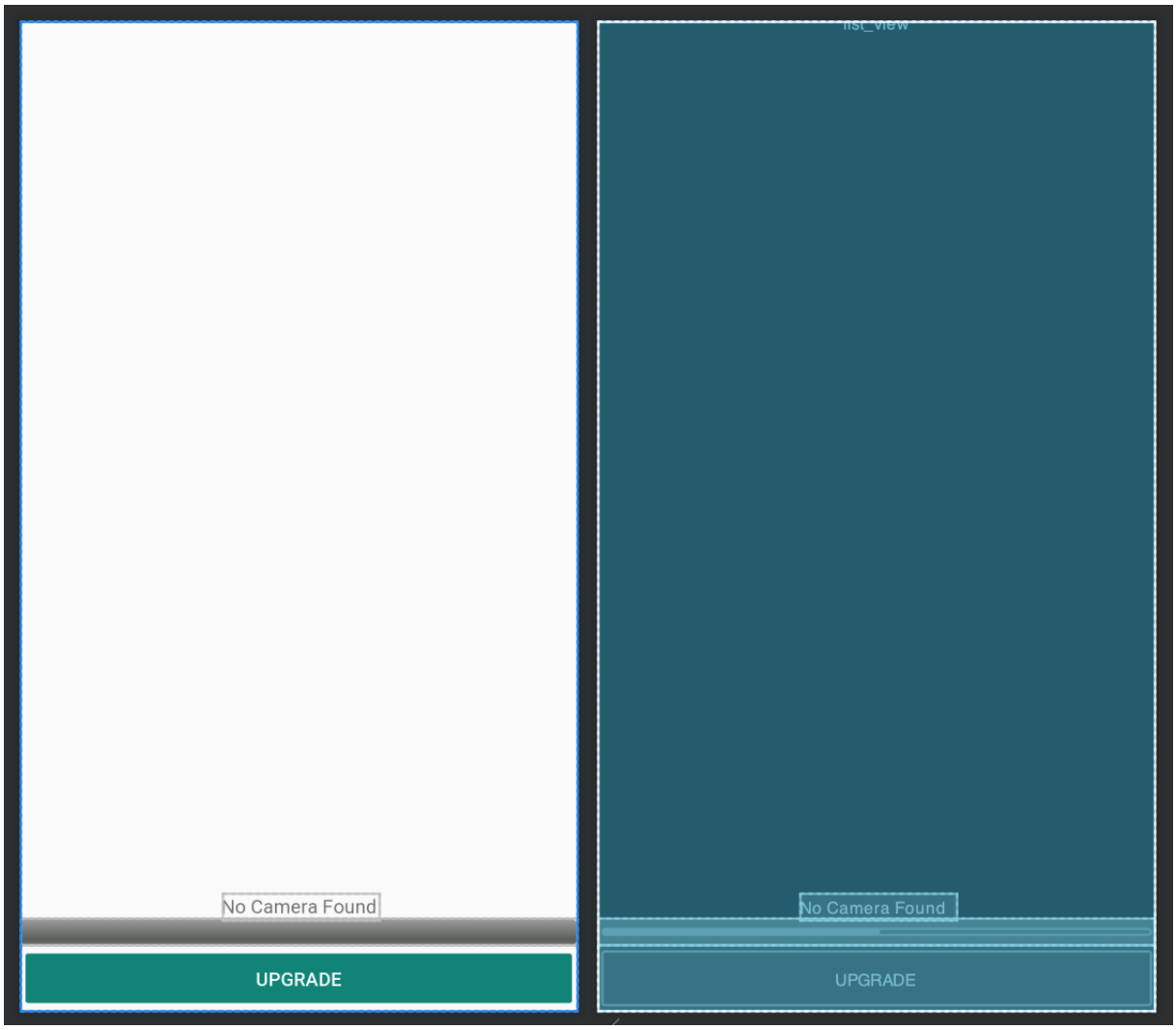


Figure 4.3 User Interface

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ListView
        android:id="@+id/list_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        />
    <TextView
        android:id="@+id/camera_info"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/no_camera_found"
        app:layout_constraintBottom_toTopOf="@id/progress_bar"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        />
    <ProgressBar
        android:id="@+id/progress_bar"
```

```
        style="@android:style/Widget.ProgressBar.Horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:progress="0"
        android:max="100"
        android:progressTint="@color/colorPrimary"
        app:layout_constraintBottom_toTopOf="@id/upgrade"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        />
    <Button
        android:id="@+id/upgrade"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/upgrade"
        android:backgroundTint="@color/colorPrimary"
        android:textColor="#ffffff"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        android:onClick="onUpgradeClicked"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Write the MainActivity.java code

Add the following handlers to the MainActivity.java source file:

```
package com.example.seekupgrade;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.os.HandlerThread;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import com.thermal.seekware.SeekCamera;
import com.thermal.seekware.SeekCameraManager;
import com.thermal.seekware.SeekUtility;
import java.util.List;
public class MainActivity extends AppCompatActivity {
    private HandlerThread handlerThread;
    private Handler handler;
    private ListView listView;
    private ProgressBar progressBar;
    private Button upgradeButton;
    private TextView cameraInfo;
    private String selectedFirmwareFile = null;
    private SeekCameraManager seekCameraManager;
    private SeekCamera seekCamera;
    private SeekCamera.StateCallback stateCallback = new SeekCamera.StateCallback() {
        @Override
        public void onInitialized(SeekCamera camera) {
        }
        @Override
        public void onOpened(SeekCamera camera) {
            seekCamera = camera;
            runOnUiThread(() -> {
                cameraInfo.setText(seekCamera.toString());
                if(selectedFirmwareFile != null){
                    upgradeButton.setVisibility(View.VISIBLE);
                }
            });
        }
        @Override
        public void onStarted(SeekCamera camera) {
        }
        @Override
        public void onStopped(SeekCamera camera) {
        }
        @Override
        public void onClosed(SeekCamera camera) {
            runOnUiThread(() -> cameraInfo.setText(getString(R.string.no_camera_found)));
        }
        @Override
        public void onMemoryAccess(SeekCamera camera, SeekCamera.MemoryRegion region, final int progress) {
            runOnUiThread(() -> progressBar.setProgress(progress));
        }
        @Override
```

```
        public void onReboot(SeekCamera seekCamera) {
            Toast.makeText(getApplicationContext(), getString(R.string.reboot), Toast.LENGTH_SHORT).show();
            runOnUiThread(() -> cameraInfo.setText(getString(R.string.reboot)));
        }
        @Override
        public void onError(SeekCamera camera, Exception e) {
            Toast.makeText(getApplicationContext(), "Firmware upgrade failed!", Toast.LENGTH_SHORT).show();
        }
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listView = findViewById(R.id.list_view);
        List<String> listFirmware = SeekUtility.findFirmwareUpgradeFiles(this, "32K");
        ArrayAdapter listAdapter = new ArrayAdapter(this, android.R.layout.simple_list_item_1,
       listFirmware);
        listView.setAdapter(listAdapter);
        listView.setOnItemClickListener((adapterView, view, i, l) -> {
            selectedFirmwareFile = adapterView.getAdapter().getItem(i).toString();
            upgradeButton.setText("UPGRADE to " + selectedFirmwareFile);
            upgradeButton.setVisibility(View.VISIBLE);
        });
        progressBar = findViewById(R.id.progress_bar);
        progressBar.setVisibility(View.INVISIBLE);
        cameraInfo = findViewById(R.id.camera_info);
        upgradeButton = findViewById(R.id.upgrade);
        upgradeButton.setVisibility(View.INVISIBLE);
        handlerThread = new HandlerThread("Background");
        handlerThread.start();
        handler = new Handler(handlerThread.getLooper());
        seekCameraManager = new SeekCameraManager(this, handler, stateCallback);
    }
    public void onUpgradeClicked(View v){
        if(seekCamera != null && selectedFirmwareFile != null){
            progressBar.setVisibility(View.VISIBLE);
            seekCamera.upgradeFirmware(selectedFirmwareFile, true);
        }
    }
}
```

Application Screenshot

Here is a screenshot of the completed application:

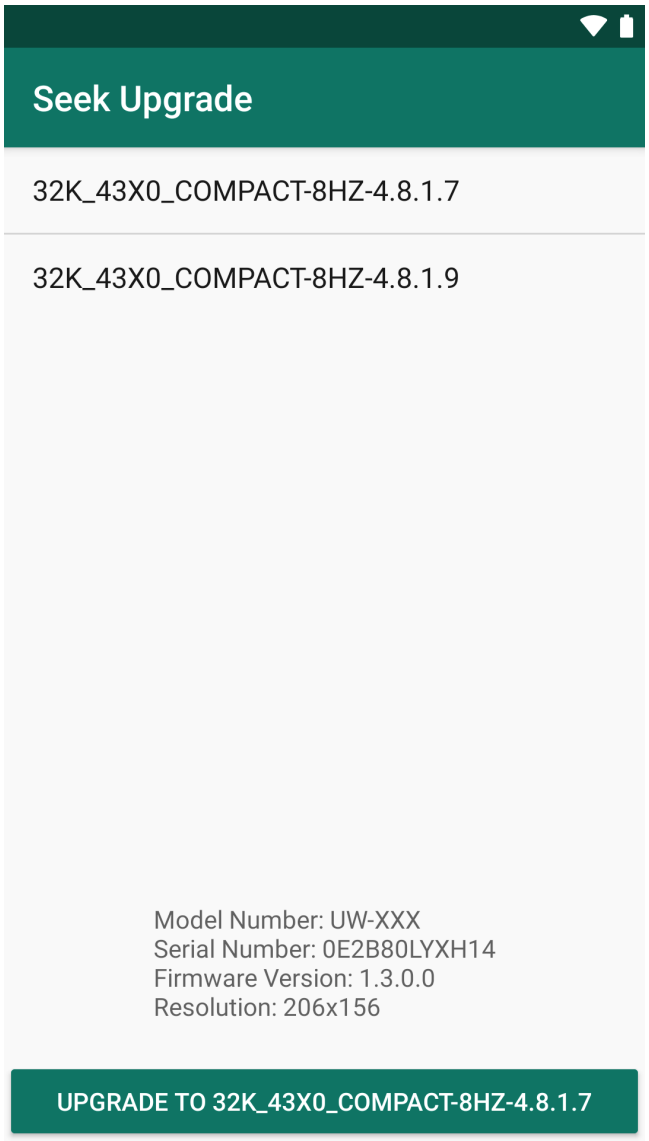


Figure 4.4 Screenshot

Chapter 5

Seek Simple Example

This example shows you how to create a Simple Thermal Camera application with Android Studio and the Seek Thermal Android SDK.

1. Create the Android Studio project
2. Change the AndroidManifest.xml file
3. Add Seek Android SDK to project
4. Create the User Interface
5. Write the MainActivity.java code
6. Application Screenshot

Create the Android Studio project

To create your new Android project, follow these steps:

1. In the Welcome to Android Studio window, click Start a new Android Studio project or if you have a project already opened, select File $>$ New $>$ New Project.
2. In the Choose your project window, select Empty Activity and click Next.
   newproject.png "New Project" width=\textwidth
3. In the Configure your project window, complete the following:

- Enter "Seek Simple" in the Name field.
- Enter "com.example.seeksimple" in the Package name field.
- If you'd like to place the project in a different folder, change its Save location.
- Select Java from the Language drop-down menu.
- Select the checkbox next to Use androidx.$*$ artifacts.
- Click Finish.

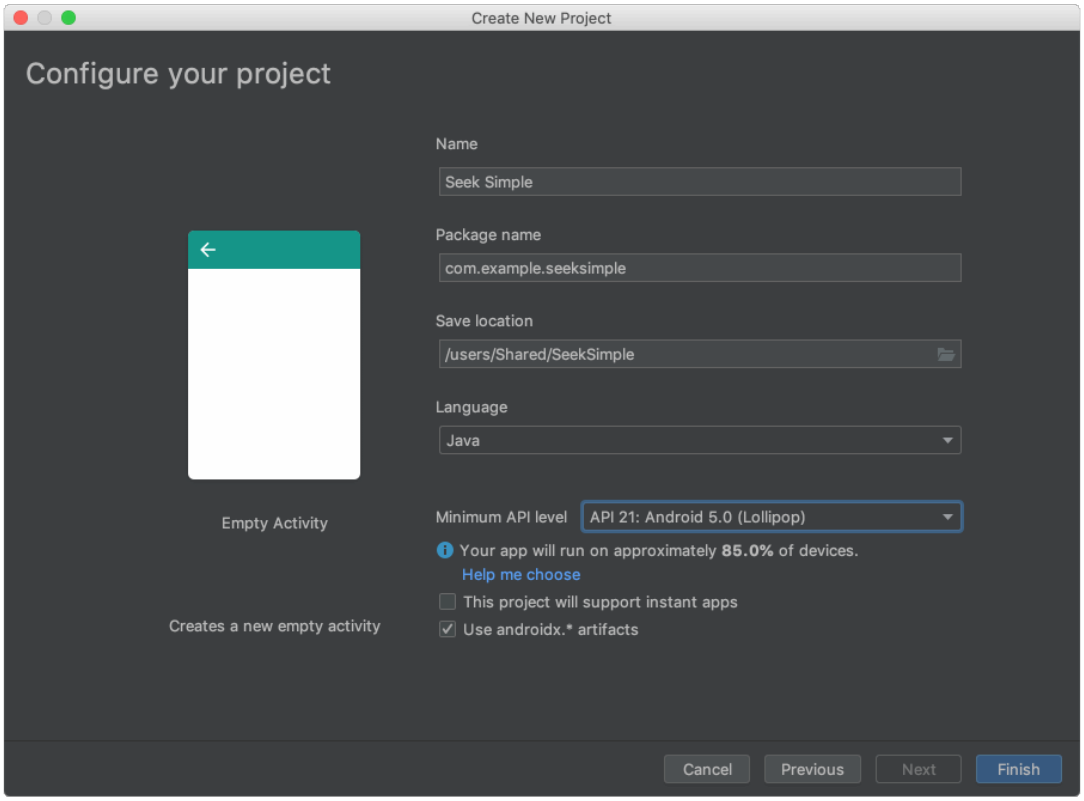


Figure 5.1 Configure Project

Change the AndroidManifest.xml file

Insert the following USB Device Intents to the AndroidManifest.xml file for ".MainActivity":

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
    </intent-filter>
    <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
        android:resource="@xml/seekware_device_filter" />
</activity>
```

Add Seek Android SDK to project

The Seekware Android 4.0 SDK is delivered as an AAR (Android Archive) for use with Android Studio and Gradle.

1. Open up the project structure by right-clicking on your project and choosing “Open Module Settings” or choosing “File” > “Project Structure...”

2. Click the “+” button in the top left to add a new module.
3. Choose “Import .JAR or .AAR Package” and click the “Next” button.
4. Find your file using the ellipsis button (“. . . ”) beside the “File name” field. Studio will automatically create a subproject name. Click “Finish”.
5. Gradle will sync, which may take a few minutes. Add the new module as a dependency to your app. In the “Project Structure...” window, a new module has appeared representing the SDK. Keep the app’s module selected and click on the Dependencies pane.
6. Use the “+” button at the bottom of the dependencies screen, and choose “module dependency”. The dependencies screen is quite tall, so this button might be invisible at first.
7. The screen that pops up should show the Seekware_Android_4.0. Click “OK”.

Create the User Interface for the Application

Create the resources for the Application using a text editor.

1. Create the res/drawable/arrow.xml resource:
```
<vector android:height="24dp" android:tint="#00FF00"
    android:viewportHeight="24.0" android:viewportWidth="24.0"
    android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#FF000000" android:pathData="M20,11H7.83l5.59,-5.59L12,4l-8,8 8,8
 1.41,-1.41L7.83,13H20v-2z"/>
</vector>
```
2. Create the res/drawable/camera.xml resource:
```
<vector android:height="32dp" android:tint="#FF0000"
    android:viewportHeight="24.0" android:viewportWidth="24.0"
    android:width="32dp" xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#FF000000" android:pathData="M12,12m-3.2,0a3.2,3.2 0,1 1,6.4 0a3.2,3.2
 0,1 1,-6.4 0"/>
    <path android:fillColor="#FF000000" android:pathData="M9,2L7.17,4L4,4c-1.1,0 -2,0.9 -2,2v12c0,1.1
 0.9,2 2,2h16c1.1,0 2,-0.9 2,-2L22,6c0,-1.1 -0.9,-2 -2,-2h-3.17L15,2L9,2zM12,17c-2.76,0 -5,-2.24
 -5,-5s2.24,-5 5,-5 5,2.24 5,5 -2.24,5 -5,5z"/>
</vector>
```
3. Create the res/drawable/ic_info.xml resource:
```
<vector android:height="50dp" android:viewportHeight="160"
    android:viewportWidth="160" android:width="50dp"
 xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#ffffff" android:pathData="m80,15c-35.88,0 -65,29.12 -65,65s29.12,65
 65,65 65,-29.12 65,-65 -29.12,-65 -65,-65zM80,25c30.36,0 55,24.64 55,55s-24.64,55 -55,55 -55,-24.64
 -55,-55 24.64,-55 55,-55z"/>
    <path android:fillColor="#ffffff" android:pathData="m89.998,51.251a11.25,11.25 89.968,1 1,-22.5
 0,11.25 11.25,89.968 1,1 22.5,0z"/>
    <path android:fillColor="#ffffff" android:pathData="m90.665,110.96c-0.069,2.73 1.211,3.5
 4.327,3.82l5.008,0.1v5.12h-39.073v-5.12l5.503,-0.1c3.291,-0.1 4.082,-1.38
 4.327,-3.82v-30.813c0.035,-4.879 -6.296,-4.113 -10.757,-3.968v-5.074l30.665,-1.105"/>
</vector>
```
4. Create the res/drawable/square.xml resource
```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="150dp"
    android:height="150dp"
    android:viewportWidth="39.687496"
    android:viewportHeight="39.687496">
  <path
      android:pathData="M2.1167,2.1139h35.4569v35.4569h-35.4569z"
      android:strokeLineJoin="round"
      android:strokeWidth="4.23333311"
      android:fillColor="#00000000"
      android:strokeColor="#0000ff"/>
</vector>
```
5. Overwrite the res/values/strings.xml resource:
```
<resources>
    <string name="app_name">Seek Simple</string>
    <string name="capture">Capture</string>
    <string name="flatfield">Flat Field</string>
    <string name="logo">Flat Field</string>
```

```
        <string name="ff">FF</string>
        <string name="prev_pal">"- Pal"</string>
        <string name="next_pal">"Pal +"</string>
        <string name="settings">Settings</string>
    </resources>
```

6. Create the Portrait User Interface layout:

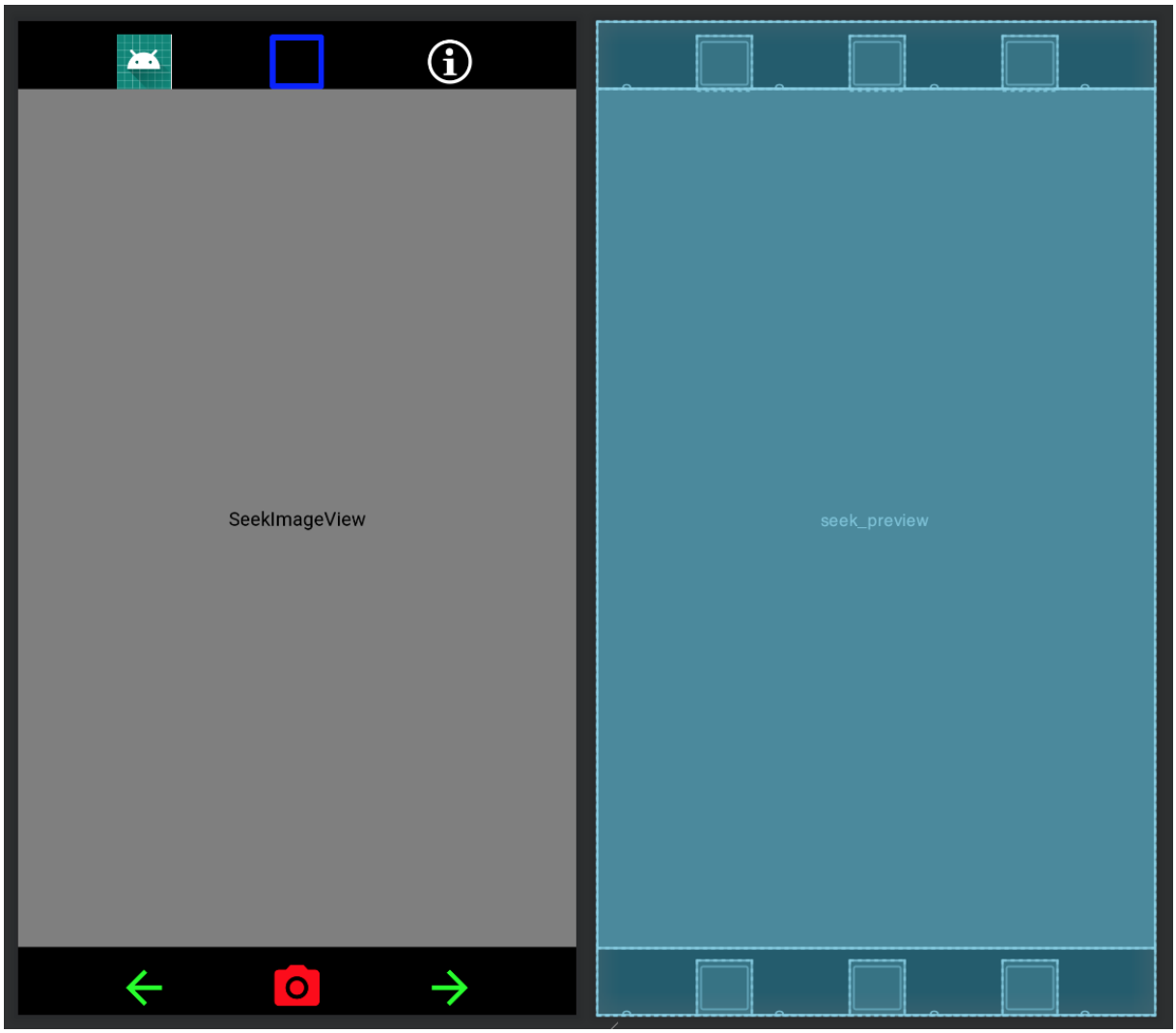


Figure 5.2 Portrait User Interface

by overwriting the res/layout/main_activity.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main_activity"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="50sp"
        android:background="#ff000000"
        android:gravity="center|top"
        android:orientation="horizontal">
        <Space
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_logo"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@mipmap/ic_launcher"
            android:contentDescription="@string/logo"
            android:visibility="visible"
            android:onClick="clickLogo" />
        <Space
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_flatfield"
            android:textColor="#ffffffff"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/square"
            android:contentDescription="@string/flatfield"
            android:onClick="clickFlatField" />
        <Space
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_info"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/ic_info"
            android:contentDescription="@string/info"
            android:onClick="clickInfo" />
        <Space
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
    </LinearLayout>
    <com.thermal.seekware.SeekImageView
        android:id="@+id/seek_preview"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="15" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="50sp"
        android:background="#ff000000"
        android:gravity="bottom|center"
        android:orientation="horizontal">
        <Space
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_prev_pal"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/arrow"
            android:contentDescription="@string/prev_pal"
            android:onClick="clickPrevPalette" />
        <Space
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_capture"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/camera"
            android:contentDescription="@string/capture"
            android:onClick="clickCapture" />
        <Space
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_next_pal"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/arrow"
            android:rotation="180"
            android:contentDescription="@string/next_pal"
            android:onClick="clickNextPalette" />
        <Space
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
            android:layout_weight="1" />
    </LinearLayout>
</LinearLayout>
```

1. Create the Landscape User Interface layout:

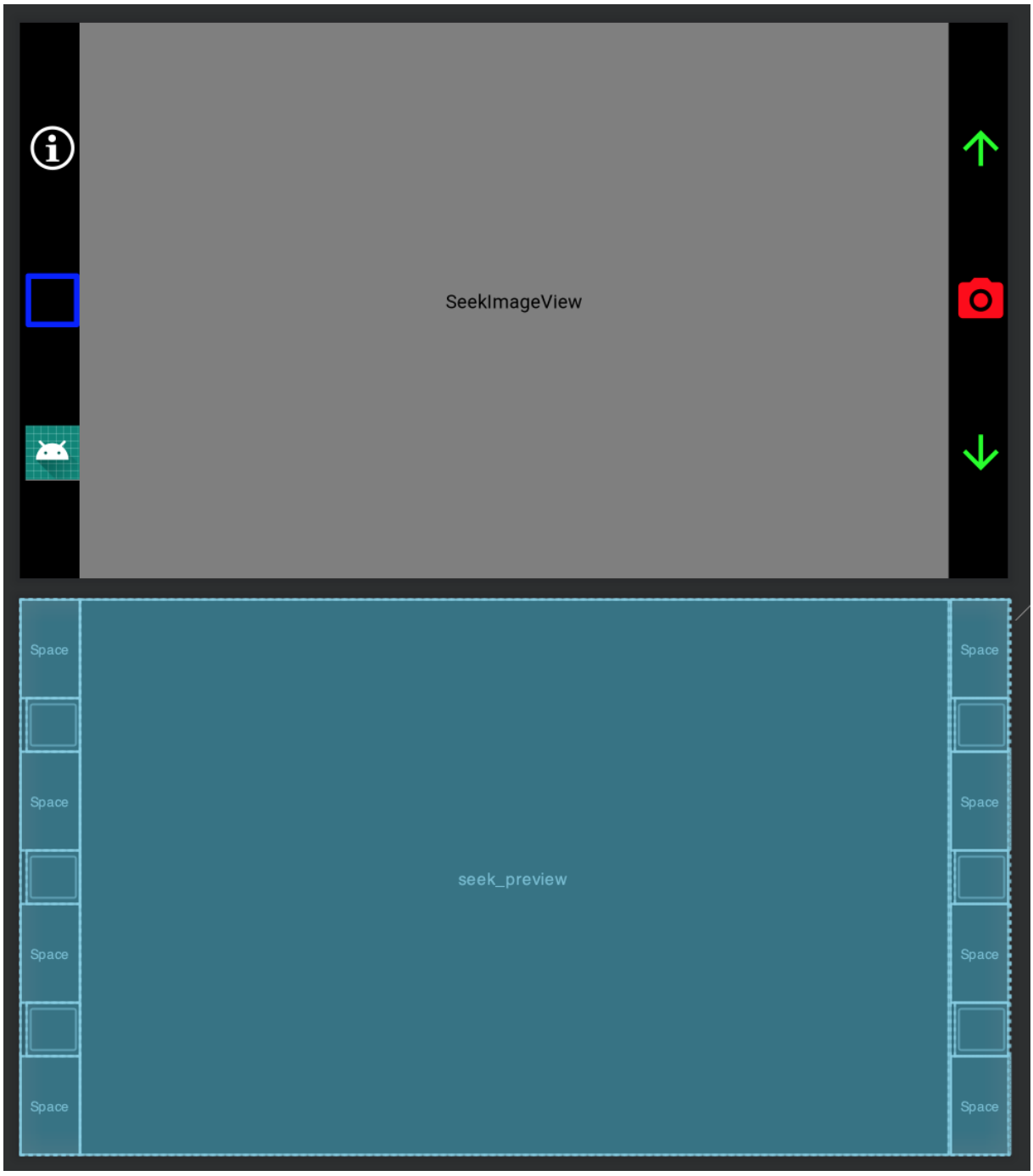


Figure 5.3 Landscape User Interface

by creating the res/layout-land/main_activity.xml in a text editor:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main_activity"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal">
    <LinearLayout
        android:layout_width="50sp"
        android:layout_height="match_parent"
        android:background="#ff000000"
        android:layout_weight="1"
        android:gravity="center|start"
        android:orientation="vertical">
        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_info"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/ic_info"
            android:contentDescription="@string/info"
            android:onClick="clickInfo"
            android:visibility="visible" />
        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_flatfield"
            android:textColor="#ffffffff"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/square"
            android:contentDescription="@string/flatfield"
            android:onClick="clickFlatField"
            android:visibility="visible" />
        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_logo"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@mipmap/ic_launcher"
            android:contentDescription="@string/logo"
            android:visibility="visible"
            android:onClick="clickLogo" />
        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
    </LinearLayout>
    <com.thermal.seekware.SeekImageView
        android:id="@+id/seek_preview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="15" />
    <LinearLayout
        android:layout_width="50sp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#ff000000"
        android:gravity="end|center"
        android:orientation="vertical">
        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_next_pal"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/arrow"
            android:rotation="90"
            android:contentDescription="@string/next_pal"
            android:onClick="clickNextPalette"
            android:visibility="visible" />
        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_capture"
```

```
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/camera"
            android:contentDescription="@string/capture"
            android:onClick="clickCapture"
            android:visibility="visible" />
        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_prev_pal"
            android:layout_width="40sp"
            android:layout_height="40sp"
            android:background="@drawable/arrow"
            android:rotation="270"
            android:contentDescription="@string/prev_pal"
            android:onClick="clickPrevPalette"
            android:visibility="visible" />
        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
    </LinearLayout>
</LinearLayout>
```

Write the MainActivity.java code

1. Create the MainActivity window. This version forces Full Screen mode and locks the screen orientation at startup in onCreate. It also overrides onFrameAvailable() and registers the callback to allow for capturing the (raw) Bitmap image.

```
package com.example.seeksimple;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.Toast;
import com.thermal.seekware.SeekCamera;
import com.thermal.seekware.SeekCameraManager;
import com.thermal.seekware.SeekImage;
import com.thermal.seekware.SeekImageView;
import com.thermal.seekware.SeekUtility;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
import java.util.Objects;
import static android.os.Environment.getExternalStorageDirectory;
public class MainActivity extends AppCompatActivity implements SeekImageView.OnFrameAvailableListener
 {
    /* Static Fields */
    private static final String TAG = MainActivity.class.getSimpleName();
    private Button flatFieldButton;
    private String captureFilename = null;
    private SeekCamera myCamera;
    private SeekCameraManager seekCameraManager;
    private SeekImageView seekPreview;
    private SeekCamera.StateCallback stateCallback = new SeekCamera.StateCallback() {
        @Override
        public void onInitialized(SeekCamera seekCamera) {
        }
        @Override
        public void onOpened(SeekCamera seekCamera) {
            myCamera = seekCamera;
            seekCamera.createSeekCameraCaptureSession(seekPreview);
        }
        @Override
        public void onStarted(SeekCamera seekCamera) {
            // Set default Color LUT
            SetPalette(0);
        }
```

```
        @Override
        public void onStopped(SeekCamera seekCamera) {
        }
        @Override
        public void onClosed(SeekCamera seekCamera) {
        }
        @Override
        public void onMemoryAccess(SeekCamera camera, SeekCamera.MemoryRegion region, final int
 progress){
        }
        @Override
        public void onReboot(SeekCamera seekCamera) {
        }
        @Override
        public void onError(SeekCamera seekCamera, Exception e) {
        }
    };
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
 int[] grantResults) {
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        //hide the title bar
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        Objects.requireNonNull(getSupportActionBar()).hide();
        super.onCreate(savedInstanceState);
        //show the activity in full screen
        Window window = getWindow();
        window.setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
 WindowManager.LayoutParams.FLAG_FULLSCREEN);
        // keep screen on
        window.addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        setContentView(R.layout.activity_main);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LOCKED);
        flatFieldButton = findViewById(R.id.btn_flatfield);
        seekPreview = findViewById(R.id.seek_preview);
        seekPreview.setOnFrameAvailableListener(this);
        seekCameraManager = new SeekCameraManager(this, null, stateCallback);
        SeekUtility.PermissionHandler.requestStoragePermission(this);
    }
```

2. Add the clickPrevPalette & clickNextPalette button handlers to MainActivity.java:

```
private int myPalette = 0;
final SeekCamera.ColorLut luts[] = {
        SeekCamera.ColorLut.TYRIAN,
        SeekCamera.ColorLut.IRON2,
        SeekCamera.ColorLut.RECON,
        SeekCamera.ColorLut.WHITEHOT,
        SeekCamera.ColorLut.BLACKHOT
};
// Change Camera Color LUT (index)
private void SetPalette(int lut) {
    myPalette = lut;
    if (myCamera != null) {
        myCamera.setColorLut(luts[lut]);
    }
}
public void clickPrevPalette(View v) {
    if (myPalette > 0) {
        SetPalette(myPalette - 1);
    } else {
        SetPalette(luts.length - 1);
    }
}
public void clickNextPalette(View v) {
    if (myPalette < luts.length - 1) {
        SetPalette(myPalette + 1);
    } else {
        SetPalette(0);
    }
}
```

3. Add the clickFlatField, clickInfo & clickLogo button handlers to MainActivity.java:

```
public void clickFlatField(View v) {
    if (myCamera != null) {
        Toast.makeText(this, "Flat Field", Toast.LENGTH_SHORT).show();
        myCamera.triggerShutter();
    }
}
public void clickInfo(View v) {
    if (myCamera != null) {
        String str = myCamera.toString();
        Toast.makeText(this, str, Toast.LENGTH_LONG).show();
    }
}
```

```
public void clickLogo(View v)
{
    String url = "http://www.thermal.com";
    Intent i = new Intent(Intent.ACTION_VIEW);
    i.setData(Uri.parse(url));
    startActivity(i);
}
```

4. Add the clickCapture command handler and spot meter display code to the MainActivity.java:

```
String captureFilename = null;
void writeImageToPNG(String filename, Bitmap bmp) {
    File strFolder = getExternalStorageDirectory();
    String pathname = strFolder + "/Pictures/" + filename + ".png";
    FileOutputStream out = null;
    try {
        out = new FileOutputStream(pathname);
        bmp.compress(Bitmap.CompressFormat.PNG, 100, out); // bmp is your Bitmap instance
        // PNG is a lossless format, the compression factor (100) is ignored
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (out != null) {
                out.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
public void clickCapture(View v) {
    if (seekPreview != null) {
        SimpleDateFormat formatter = new SimpleDateFormat("yyyyMMdd_HHmmss", Locale.US);
        Date now = new Date();
        captureFilename = "IR_" + formatter.format(now);
    }
}
// https://stackoverflow.com/questions/11140285/how-do-we-use-runonuithread-in-android
private void updateSpot(final float spotTemp)
{
    MainActivity.this.runOnUiThread(new Runnable() {
        public void run() {
            showSpot(spotTemp);
        }
    });
}
private void showSpot(final float spot)
{
    String spotStr = String.format("%.0f", spot);
    Button btnFlatField = (Button) this.findViewById(R.id.btn_flatfield);
    if (btnFlatField != null) {
        btnFlatField.setText(spotStr);
    }
}
@Override
public void onFrameAvailable(SeekImageView seekPreview, SeekImage seekImage) {
    SeekUtility.Temperature spotTemp = seekImage.getThermography().getSpotTemp();
    if (spotTemp != null) {
        updateSpot(spotTemp.getValue());
    }
    if (captureFilename != null) {
        // TODO: Add rotation to store image upright with USB Compact cameras on the bottom
        Toast.makeText(this, "Capture:" + captureFilename, Toast.LENGTH_SHORT).show();
        writeImageToPNG(captureFilename, seekImage.getColorBitmap());
        captureFilename = null;
    }
}
```

5. Create the res/layout/firmware_upgrade_alert.xml resource in a text editor:

```
 {(.xml)}
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ListView
        android:id="@+id/firmware_upgrade_bundle_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </ListView>
    <ProgressBar
        android:id="@+id/firmware_upgrade_progress"
        style="@android:style/Widget.ProgressBar.Horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:progress="0"
```

```
            android:max="100"
            android:progressTint="@color/colorPrimary">
        </ProgressBar>
        <TextView
            android:id="@+id/firmware_upgrade_progress_text"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:textAlignment="center">
        </TextView>
    </LinearLayout>
```

Application Screenshot

Here is a screenshot of the completed application:

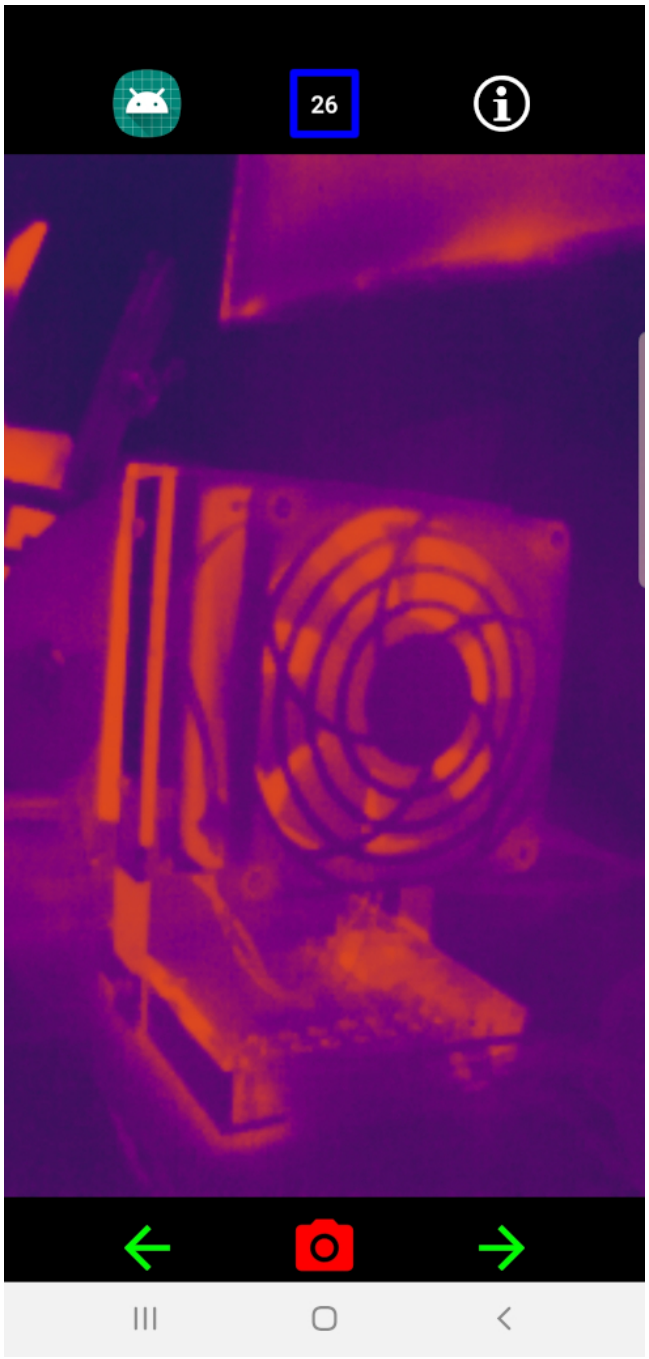


Figure 5.4 Screenshot

Chapter 6

Hierarchical Index

6.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 7

Class Index

7.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 8

Class Documentation

8.1 com.thermal.seekware.SeekCamera.AGCMode Enum Reference

Automatic gain control (AGC) mode.

Public Attributes

- LEGACY_HISTEQ
- LINEAR
- HISTEQ

8.1.1 Detailed Description

Automatic gain control (AGC) mode.

The automatic gain control (AGC) mode used by the camera

8.1.2 Member Data Documentation

8.1.2.1 HISTEQ

`com.thermal.seekware.SeekCamera.AGCMode.HISTEQ`

Default mode, uses histogram equalization Histogram Equalization

8.1.2.2 LEGACY_HISTEQ

`com.thermal.seekware.SeekCamera.AGCMode.LEGACY_HISTEQ`

Deprecated version of histogram equalization Legacy Histogram Equalization

8.1.2.3 LINEAR

`com.thermal.seekware.SeekCamera.AGCMode.LINEAR`

Linearly stretches the color bar to fit the range of temperatures Linear

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.2 com.thermal.seekware.SeekCamera.AspectRatio Enum Reference

Controls the stretching of the image.

Public Attributes

- MATCH_WIDTH
- MATCH_HEIGHT
- AUTO

8.2.1 Detailed Description

Controls the stretching of the image.

The aspect ratio of the image will always be preserved (it will never be distorted). The actual aspect ratio of the display depends on the container that it is filling (to constrain a View to a specific ratio, you can use a ConstraintLayout). This describes how it fills the screen. In the images below, the areas within the green borders represents the screen, with the blue parts representing the image, and the black parts representing blank areas that result from the aspect ratio difference.

8.2.2 Member Data Documentation

8.2.2.1 AUTO

`com.thermal.seekware.SeekCamera.AspectRatio.AUTO`

This will act like MATCH_WIDTH when the screen aspect ratio is less than the camera's, and MATCH_HEIGHT when it is greater. There will never be any black bars in this mode, only portions of the image cut off. This is the default mode.

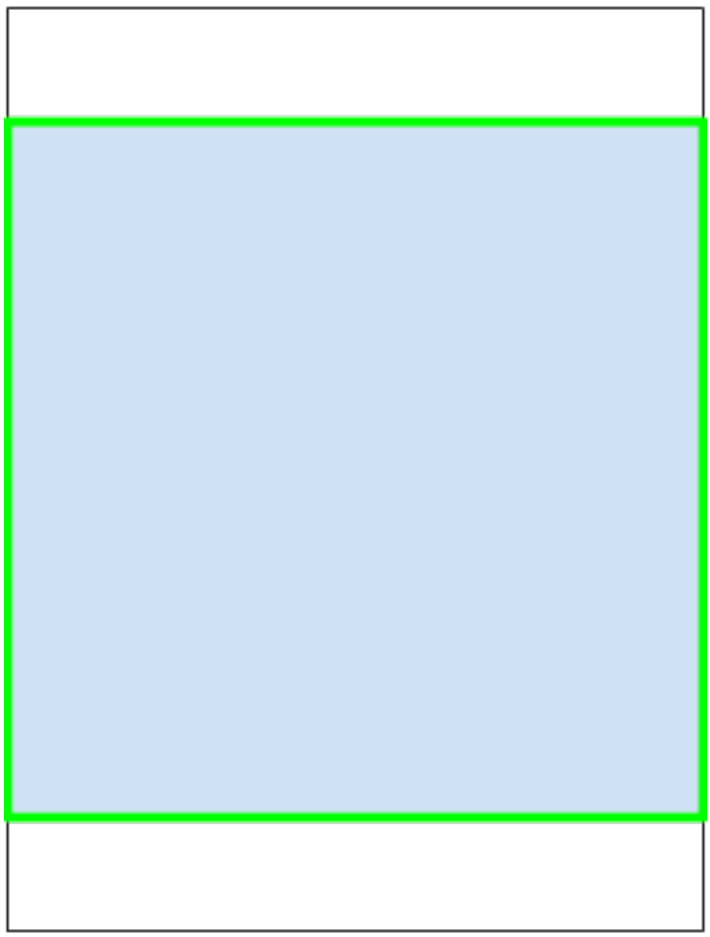


Figure 8.1 Square Portrait

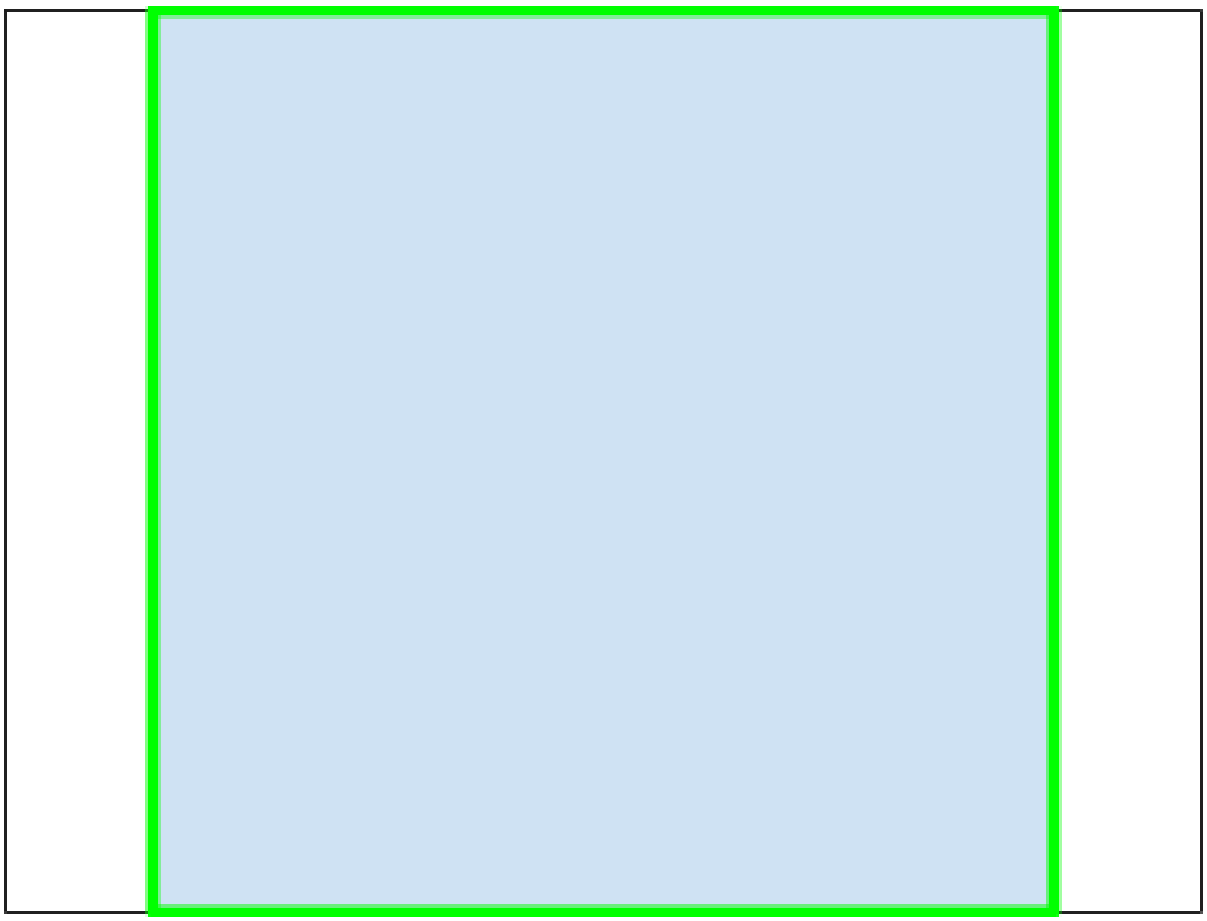


Figure 8.3 Square Landscape

8.2.2.2 MATCH_HEIGHT

`com.thermal.seekware.SeekCamera.AspectRatio.MATCH_HEIGHT`

This will stretch the image to match the height of the screen. If the screen aspect ratio is greater than the camera's, the left and right will be cut off. If it less, then there will be black bars on the sides.

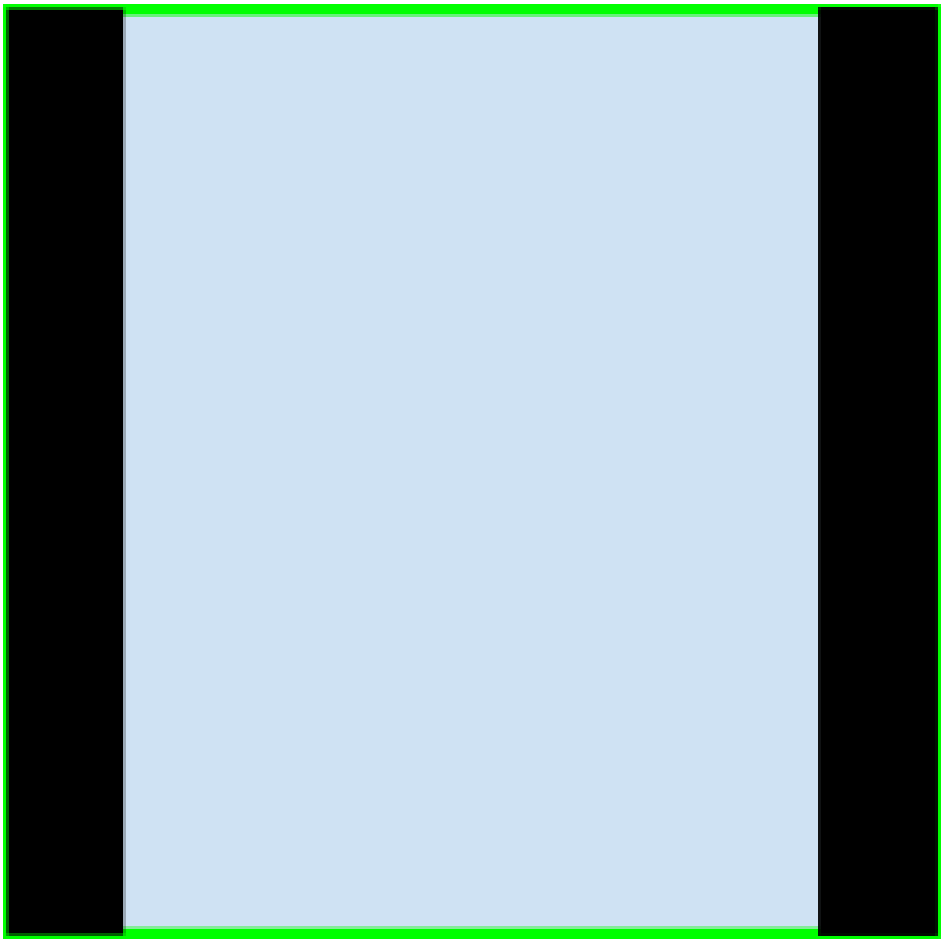


Figure 8.5 Square Portrait

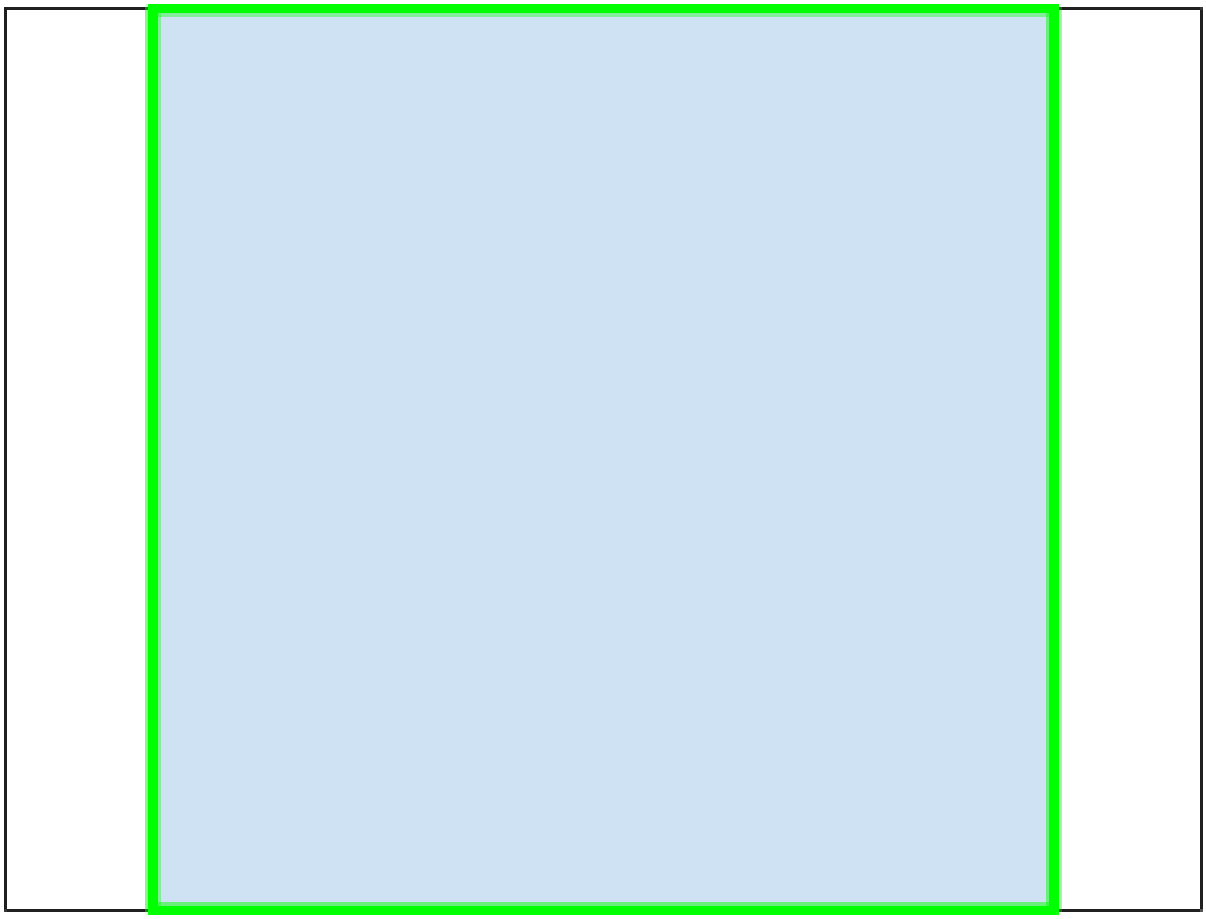


Figure 8.7 Square Landscape

8.2.2.3 MATCH_WIDTH

`com.thermal.seekware.SeekCamera.AspectRatio.MATCH_WIDTH`

This will stretch the image to match the width of the screen. If the screen aspect ratio is greater than the camera's, there will be black bars on the top and bottom. If it less, then the top and bottom will be cut off.

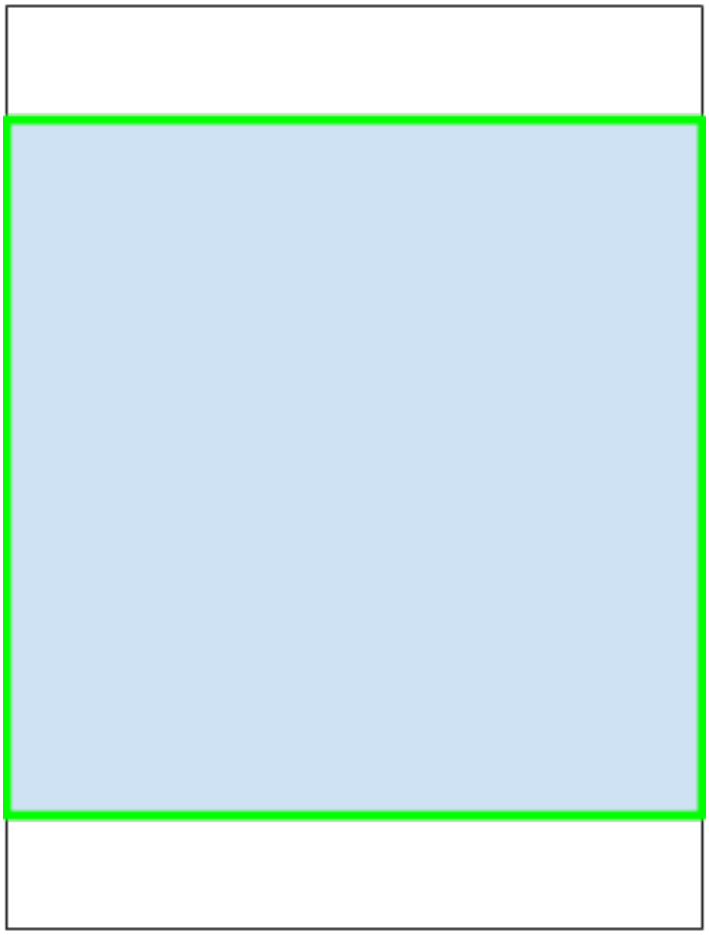


Figure 8.9 Square Portrait

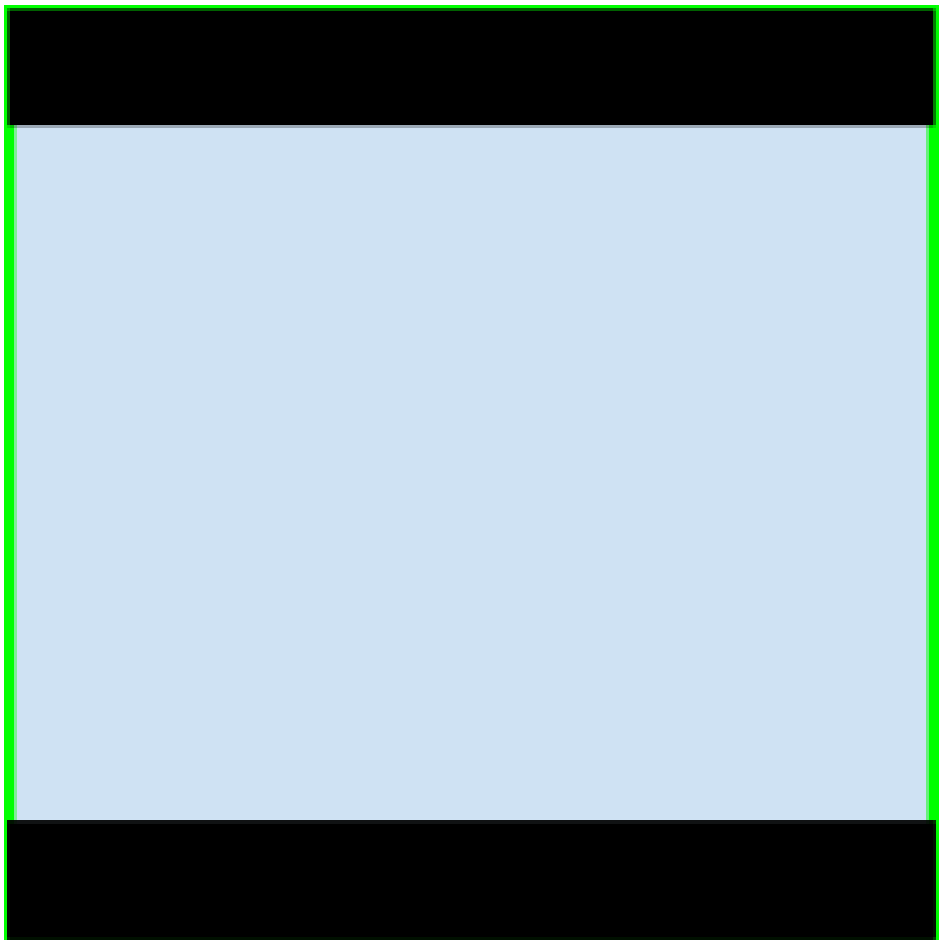


Figure 8.11 Square Landscape

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.3 com.thermal.seekware.SeekCamera.Characteristics Class Reference

Holds information about the SeekCamera.

Public Member Functions

- int getWidth ()
- int getHeight ()
- SeekUtility.Temperature.Unit getTemperatureUnit ()
- void setTemperatureUnit (@NonNull SeekUtility.Temperature.Unit temperatureUnit)
- String getSerialNumber ()
- String getModelNumber ()
- String getFirmwareVersion ()

- LensFacing getLensFacing ()
- void setLensFacing (@NonNull LensFacing lensFacing)
- Orientation getOrientation ()
- void setOrientation (@NonNull Orientation orientation)
- int getCorrectedOrientation ()
- int getSensorOrientation ()
- Point convertScreenToSensorPoint (@NonNull Point point, @NonNull Size screenSize)
- Point convertSensorToScreenPoint (@NonNull Point point, @NonNull Size screenSize)

8.3.1 Detailed Description

Holds information about the SeekCamera.

Simple struct-like class that holds information associated with a SeekCamera.

8.3.2 Member Function Documentation

8.3.2.1 convertScreenToSensorPoint()

```
Point com.thermal.seekware.SeekCamera.Characteristics.convertScreenToSensorPoint (
            @NonNull Point point,
            @NonNull Size screenSize )
```

Rotate screen coordinates to sensor coordinates

Parameters

point	Point (X,Y) in screen coordinates to rotate
screenSize	width, height of screen

Returns

Point in FPA coordinates

8.3.2.2 convertSensorToScreenPoint()

```
Point com.thermal.seekware.SeekCamera.Characteristics.convertSensorToScreenPoint (
            @NonNull Point point,
            @NonNull Size screenSize )
```

Rotate min/max coordinates to screen coordinates based on sensorOrientation

Parameters

point	Point (X,Y) in FPA coordinates to rotate
screenSize	width, height of screen

Returns

Point in logical sensor coordinates

8.3.2.3 getCorrectedOrientation()

```
int com.thermal.seekware.SeekCamera.Characteristics.getCorrectedOrientation ( )
```

Gets the corrected orientation based on orientation and sensor orientation

Returns

the corrected orientation

8.3.2.4 getFirmwareVersion()

```
String com.thermal.seekware.SeekCamera.Characteristics.getFirmwareVersion ( )
```

Gets the firmware version of this SeekCamera. This is formatted as in the form 1.2.3.4, where

1. represents the major version number
2. represents the minor version number
3. represents the major build number
4. represents the minor build number

Returns

the firmware version

8.3.2.5 getHeight()

```
int com.thermal.seekware.SeekCamera.Characteristics.getHeight ( )
```

Gets the native sensor height of an image frame in pixels. This value is always less than getWidth().

Returns

the native sensor height of an image frame

8.3.2.6 getLensFacing()

```
LensFacing com.thermal.seekware.SeekCamera.Characteristics.getLensFacing ( )
```

Gets the current direction the lens is facing. Default is LensFacing#BACK.

Returns

the current direction the lens is facing

8.3.2.7 getModelNumber()

```
String com.thermal.seekware.SeekCamera.Characteristics.getModelNumber ( )
```

Gets the model number of this SeekCamera

Returns

the model number

8.3.2.8 getOrientation()

```
Orientation com.thermal.seekware.SeekCamera.Characteristics.getOrientation ( )
```

Gets the current orientation. Default is Orientation#ORIENTATION_0.

Returns

the orientation

8.3.2.9 getSensorOrientation()

```
int com.thermal.seekware.SeekCamera.Characteristics.getSensorOrientation ( )
```

Gets the sensor orientation.

Returns

the orientation in degrees

8.3.2.10 getSerialNumber()

```
String com.thermal.seekware.SeekCamera.Characteristics.getSerialNumber ( )
```

Gets the serial number of this SeekCamera

Returns

the serial number

8.3.2.11 getTemperatureUnit()

```
SeekUtility.Temperature.Unit com.thermal.seekware.SeekCamera.Characteristics.getTemperature↩
Unit ( )
```

Gets the current temperature units used by the camera. Default is Celsius.

Returns

the current temperature unit

8.3.2.12 getWidth()

```
int com.thermal.seekware.SeekCamera.Characteristics.getWidth ( )
```

Gets the native sensor width of an image frame in pixels from the camera. This value is always more than getHeight().

Returns

the native sensor width of an image frame

8.3.2.13 setLensFacing()

```
void com.thermal.seekware.SeekCamera.Characteristics.setLensFacing (
            @NonNull LensFacing lensFacing )
```

Sets the direction the lens is facing.

Parameters

lensFacing	the desired direction the lens is facing

8.3.2.14 setOrientation()

```
void com.thermal.seekware.SeekCamera.Characteristics.setOrientation (
            @NonNull Orientation orientation )
```

Sets the orientation to the given Orientation

Parameters

orientation	the desired orientation

8.3.2.15 setTemperatureUnit()

```
void com.thermal.seekware.SeekCamera.Characteristics.setTemperatureUnit (
            @NonNull SeekUtility.Temperature.Unit temperatureUnit )
```

Sets the temperature units used by the camera to the given unit.

Parameters

temperatureUnit	the desired temperature unit

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.4 com.thermal.seekware.SeekCamera.ColorLut Enum Reference

Color Look Up Table (LUT)

Static Public Member Functions

- static byte[ ] createUserLut (@ColorInt int startColor, @ColorInt int endColor)

Public Attributes

- WHITEHOT =(100)
- BLACKHOT =(101)
- SPECTRA =(102)
- PRISM =(103)
- TYRIAN =(104)

- IRON =(105)
- AMBER =(106)
- HI =(107)
- HILO =(108)
- IRON2 =(109)
- GREEN =(110)
- RECON =(111)
- BLACK_RECON =(112)
- USER0 =(10000)
- USER1 =(10001)
- USER2 =(10002)
- USER3 =(10003)
- USER4 =(10004)

8.4.1 Detailed Description

Color Look Up Table (LUT)

Defines the list of color LUTs that can be applied to display video frames.

8.4.2 Member Function Documentation

8.4.2.1 createUserLut()

```
static byte [] com.thermal.seekware.SeekCamera.ColorLut.createUserLut (
            @ColorInt int startColor,
            @ColorInt int endColor )  [static]
```

Creates a user lut with 256 colors that can be used with loadUserLutData

Parameters

startColor	the color at the beginning of the array
endColor	the color at the end of the array

Returns

a user lut with the gradient blended between the start and end color typically used with SeekCamera.loadUserLutData

8.4.3 Member Data Documentation

8.4.3.1 AMBER

com.thermal.seekware.SeekCamera.ColorLut.AMBER =(106)

Amber LUT

8.4.3.2 BLACK_RECON

com.thermal.seekware.SeekCamera.ColorLut.BLACK_RECON =(112)

Black Recon LUT TODO: add images

8.4.3.3 BLACKHOT

com.thermal.seekware.SeekCamera.ColorLut.BLACKHOT =(101)

Black Hot LUT

8.4.3.4 GREEN

com.thermal.seekware.SeekCamera.ColorLut.GREEN =(110)

Green LUT

8.4.3.5 HI

`com.thermal.seekware.SeekCamera.ColorLut.HI =(107)`

Hi LUT

8.4.3.6 HILO

`com.thermal.seekware.SeekCamera.ColorLut.HILO =(108)`

HiLo LUT

8.4.3.7 IRON

`com.thermal.seekware.SeekCamera.ColorLut.IRON =(105)`

Iron LUT

8.4.3.8 IRON2

`com.thermal.seekware.SeekCamera.ColorLut.IRON2 =(109)`

Iron2 LUT

8.4.3.9 PRISM

`com.thermal.seekware.SeekCamera.ColorLut.PRISM =(103)`

Prism LUT

8.4.3.10 RECON

`com.thermal.seekware.SeekCamera.ColorLut.RECON =(111)`

Recon LUT

8.4.3.11 SPECTRA

`com.thermal.seekware.SeekCamera.ColorLut.SPECTRA =(102)`

Spectra LUT

8.4.3.12 TYRIAN

`com.thermal.seekware.SeekCamera.ColorLut.TYRIAN =(104)`

Tyrian LUT

8.4.3.13 USER0

```
com.thermal.seekware.SeekCamera.ColorLut.USER0 =(10000)
```

User LUT #0

See also

Must be loaded first using loadUserLutData

8.4.3.14 USER1

```
com.thermal.seekware.SeekCamera.ColorLut.USER1 =(10001)
```

User LUT #1

See also

Must be loaded first using loadUserLutData

8.4.3.15 USER2

```
com.thermal.seekware.SeekCamera.ColorLut.USER2 =(10002)
```

User LUT #2

See also

Must be loaded first using loadUserLutData

8.4.3.16 USER3

```
com.thermal.seekware.SeekCamera.ColorLut.USER3 =(10003)
```

User LUT #3

See also

Must be loaded first using loadUserLutData

8.4.3.17 USER4

```
com.thermal.seekware.SeekCamera.ColorLut.USER4 =(10004)
```

User LUT #4

See also

Must be loaded first using loadUserLutData

8.4.3.18 WHITEHOT

```
com.thermal.seekware.SeekCamera.ColorLut.WHITEHOT =(100)
```

White Hot LUT

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.5 com.thermal.seekware.SeekIOException.ErrorCode Enum Reference

Error codes that describe communication failures between the SDK and a SeekCamera.

Public Member Functions

- int value ()

Static Public Member Functions

- static boolean contains (int value)
- static ErrorCode get (int value)

8.5.1 Detailed Description

Error codes that describe communication failures between the SDK and a SeekCamera.

This enum contains error codes associated with IO errors in the SDK

8.5.2 Member Function Documentation

8.5.2.1 contains()

```
static boolean com.thermal.seekware.SeekIOException.ErrorCode.contains (
            int value )  [static]
```

Returns true if the given value is in the enum

Parameters

value	the desired value

Returns

whether or not the value is in the enum

8.5.2.2 get()

```
static ErrorCode com.thermal.seekware.SeekIOException.ErrorCode.get (
            int value )  [static]
```

Gets the code for the given value and returns it

Parameters

value	the desired value

Returns

the code for the given value

8.5.2.3 value()

```
int com.thermal.seekware.SeekIOException.ErrorCode.value ( )
```

Getter for this.value

Returns

the error code of this enum

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekIOException.java

8.6 com.thermal.seekware.SeekPipelineException.ErrorCode Enum Reference

Custom error codes.

Public Member Functions

- int value ()

Static Public Member Functions

- static boolean contains (int value)
- static ErrorCode get (int value)

8.6.1 Detailed Description

Custom error codes.

This enum contains error codes associated with all types of errors in the SDK

8.6.2 Member Function Documentation

8.6.2.1 contains()

```
static boolean com.thermal.seekware.SeekPipelineException.ErrorCode.contains (
            int value )  [static]
```

Returns true if the given value is in the enum

Parameters

value	the desired value

Returns

whether or not the value is in the enum

8.6.2.2 get()

```
static ErrorCode com.thermal.seekware.SeekPipelineException.ErrorCode.get (
            int value )  [static]
```

Gets the code for the given value and returns it

Parameters

value	the desired value

Returns

the code for the given value

8.6.2.3 value()

```
int com.thermal.seekware.SeekPipelineException.ErrorCode.value ( )
```

Getter for this.value

Returns

the error code of this enum

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekPipelineException.java

8.7 com.thermal.seekware.SeekCamera.LensFacing Enum Reference

Direction the camera lens is facing.

Public Attributes

- FRONT
- BACK

8.7.1 Detailed Description

Direction the camera lens is facing.

Which direction the camera lens is facing

8.7.2 Member Data Documentation

8.7.2.1 BACK

```
com.thermal.seekware.SeekCamera.LensFacing.BACK
```

traditional back facing camera Back facing

8.7.2.2 FRONT

`com.thermal.seekware.SeekCamera.LensFacing.FRONT`

selfie camera, flips only the preview horizontally, the media capture will be in the sensor orientation Front facing (selfie)

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.8 com.thermal.seekware.SeekCamera.MemoryRegion Enum Reference

Specifies the region of firmware memory to read or write.

Public Attributes

- NEW_IMAGE
- IMAGE0
- IMAGE1

8.8.1 Detailed Description

Specifies the region of firmware memory to read or write.

8.8.2 Member Data Documentation

8.8.2.1 IMAGE0

`com.thermal.seekware.SeekCamera.MemoryRegion.IMAGE0`

The region of firmware memory used to store the first firmware image.

8.8.2.2 IMAGE1

`com.thermal.seekware.SeekCamera.MemoryRegion.IMAGE1`

The region of firmware memory used to store the second firmware image.

8.8.2.3 NEW_IMAGE

`com.thermal.seekware.SeekCamera.MemoryRegion.NEW_IMAGE`

The region of firmware memory used to store a new firmware image.

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.9 com.thermal.seekware.Thermography.Metrics Class Reference

Simple struct to hold average, min, and max temperatures.

Public Member Functions

- SeekUtility.Temperature getAverage ()
- SeekUtility.Temperature getMin ()
- SeekUtility.Temperature getMax ()

8.9.1 Detailed Description

Simple struct to hold average, min, and max temperatures.

8.9.2 Member Function Documentation

8.9.2.1 getAverage()

`SeekUtility.Temperature com.thermal.seekware.Thermography.Metrics.getAverage ( )`

Gets the average temperature

Returns

the average temperature

8.9.2.2 getMax()

```
SeekUtility.Temperature com.thermal.seekware.Thermography.Metrics.getMax ( )
```

Gets the max temperature

Returns

the max temperature

8.9.2.3 getMin()

```
SeekUtility.Temperature com.thermal.seekware.Thermography.Metrics.getMin ( )
```

Gets the min temperature

Returns

the min temperature

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/Thermography.java

8.10 com.thermal.seekware.SeekImageView.OnFrameAvailableListener Interface Reference

Listens for an available frame.

Public Member Functions

- void onFrameAvailable (SeekImageView seekImageView, SeekImage seekImage)

8.10.1 Detailed Description

Listens for an available frame.

8.10.2 Member Function Documentation

8.10.2.1 onFrameAvailable()

```
void com.thermal.seekware.SeekImageView.OnFrameAvailableListener.onFrameAvailable (
            SeekImageView seekImageView,
            SeekImage seekImage )
```

Called when a frame is available

Parameters

seekImageView	SeekImageView associated with callback
seekImage	SeekImage associated with callback

The documentation for this interface was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekImageView.java

8.11 com.thermal.seekware.SeekImageReader.OnImageAvailableListener Interface Reference

Called when a SeekImage is available.

Public Member Functions

- void onImageAvailable (SeekImage seekImage)

8.11.1 Detailed Description

Called when a SeekImage is available.

Listens for when a SeekImage is available

8.11.2 Member Function Documentation

8.11.2.1 onImageAvailable()

```
void com.thermal.seekware.SeekImageReader.OnImageAvailableListener.onImageAvailable (
            SeekImage seekImage )
```

Called once the pipeline is fully completed one cycle and has an image ready for use

Parameters

seekImage	the SeekImage associated with the callback

The documentation for this interface was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekImageReader.java

8.12 com.thermal.seekware.SeekCamera.Orientation Enum Reference

Orientation used for image rendering.

Public Member Functions

- int getCorrectedOrientation (int sensorOrientation)
- Orientation next ()
- Orientation previous ()

Public Attributes

- ORIENTATION_0 =(0)
- ORIENTATION_90 =(90)
- ORIENTATION_180 =(180)
- ORIENTATION_270 =(270)

8.12.1 Detailed Description

Orientation used for image rendering.

Refers to the orientation in which to render the image

8.12.2 Member Function Documentation

8.12.2.1 getCorrectedOrientation()

```
int com.thermal.seekware.SeekCamera.Orientation.getCorrectedOrientation (
            int sensorOrientation )
```

Gets the corrected orientation (adjusted for SPI and USB native sensor orientation). Note: this value will always be clamped between 0 and 360.

Parameters

sensorOrientation	the sensor orientation from SeekCameraConnection

Returns

the corrected orientation in degrees (between 0 and 360)

8.12.2.2 next()

```
Orientation com.thermal.seekware.SeekCamera.Orientation.next ( )
```

Gets the next Orientation

Returns

the next Orientation in the sequence

8.12.2.3 previous()

```
Orientation com.thermal.seekware.SeekCamera.Orientation.previous ( )
```

Gets the previous Orientation

Returns

the previous Orientation in the sequence

8.12.3 Member Data Documentation

8.12.3.1 ORIENTATION_0

```
com.thermal.seekware.SeekCamera.Orientation.ORIENTATION_0 =(0)
```

Orientation 0 Degrees

8.12.3.2 ORIENTATION_180

```
com.thermal.seekware.SeekCamera.Orientation.ORIENTATION_180 =(180)
```

Orientation 180 Degrees

8.12.3.3 ORIENTATION_270

```
com.thermal.seekware.SeekCamera.Orientation.ORIENTATION_270 =(270)
```

Orientation 270 Degrees

8.12.3.4 ORIENTATION_90

```
com.thermal.seekware.SeekCamera.Orientation.ORIENTATION_90 =(90)
```

Orientation 90 Degrees

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.13 com.thermal.seekware.SeekUtility.OrientationManager Class Reference

Manages orientation of the devices and rotation of views and overlays.

Inherits LifecycleObserver.

Public Member Functions

- int getOrientation ()
- int getRawOrientation ()
- void addViews (View... views)

8.13.1 Detailed Description

Manages orientation of the devices and rotation of views and overlays.

This class handles the orientation as well as rotating views that should be rotated with the screen orientation.

8.13.2 Member Function Documentation

8.13.2.1 addViews()

```
void com.thermal.seekware.SeekUtility.OrientationManager.addViews (
            View...  views )
```

Adds any numbers of views that should be rotated with the screen orientation

8.13.2.2 getOrientation()

```
int com.thermal.seekware.SeekUtility.OrientationManager.getOrientation ( )
```

Gets the orientation for saving images and returns it

Returns

the orientation in degrees

8.13.2.3 getRawOrientation()

```
int com.thermal.seekware.SeekUtility.OrientationManager.getRawOrientation ( )
```

Gets the raw orientation and returns it

Returns

the raw orientation in degrees

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekUtility.java

8.14 com.thermal.seekware.SeekUtility.PermissionHandler.Permission Enum Reference

Represents a permission request.

8.14.1 Detailed Description

Represents a permission request.

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekUtility.java

8.15 com.thermal.seekware.SeekUtility.PermissionHandler Class Reference

Handles Android 6+ permissions.

Classes

- enum Permission

  Represents a permission request.
- interface StateCallback

  Callback for when a permission is granted.

8.15.1 Detailed Description

Handles Android 6+ permissions.

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekUtility.java

8.16 com.thermal.seekware.SeekCamera Class Reference

The main control for the pipeline.

Inherits LifecycleObserver.

Classes

- enum AGCMode

  Automatic gain control (AGC) mode.
- enum AspectRatio

  Controls the stretching of the image.
- class Characteristics

  Holds information about the SeekCamera.
- enum ColorLut

  Color Look Up Table (LUT)
- enum LensFacing

  Direction the camera lens is facing.
- enum MemoryRegion

  Specifies the region of firmware memory to read or write.
- enum Orientation

  Orientation used for image rendering.
- interface SeekExceptionListener

  Listens for SeekIOExceptions.
- enum State

  The current state of the camera.
- interface StateCallback

  Allows for callbacks to the states of the camera.

Public Member Functions

- AspectRatio getAspectRatio ()
- void setAspectRatio (@NonNull AspectRatio aspectRatio)
- float getEmissivity ()
- void setEmissivity (float emissivity)
- boolean getImageSmoothing ()
- void setImageSmoothing (boolean imageSmoothing)
- ColorLut getColorLut ()
- void setColorLut (@NonNull ColorLut colorLut)
- String getColorLutName ()
- void setSeekExceptionListener (SeekExceptionListener seekExceptionListener)
- Characteristics getCharacteristics ()
- ByteBuffer getColorPalette ()
- ByteBuffer getColorPalette (ColorLut colorLut)
- State getCurrentState ()
- String getChipId ()
- void triggerShutter ()
- void suspendShutter ()
- void resumeShutter ()
- void loadUserLutData (byte[ ] userLutData, ColorLut userLut)
- void setAGCMode (@NonNull AGCMode agcMode)
- synchronized void open ()
- synchronized void start ()
- synchronized void stop ()
- synchronized void close ()
- synchronized void memoryRead (MemoryRegion region, byte[ ] data) throws UnsupportedOperation↩ Exception
- synchronized void memoryRead (MemoryRegion region, int regionOffset, byte[ ] data) throws Unsupported↩ OperationException
- synchronized void memoryWrite (MemoryRegion region, int regionOffset, byte[ ] data)
- synchronized void memoryWrite (MemoryRegion region, byte[ ] data)
- void upgradeFirmware (String firmwareFilename, boolean rebootAfterUpgrade)
- void upgradeFirmware (String firmwareFilename, boolean rebootAfterUpgrade, int rebootMsDelay)
- synchronized void reboot (int msDelay)
- void createSeekCameraCaptureSession (boolean getFiltered, boolean getTherm, boolean getColor, Seek↩ PipelineListener... listeners)
- void createSeekCameraCaptureSession (SeekPipelineListener... listeners)
- String toString ()

8.16.1 Detailed Description

The main control for the pipeline.

Overview

Controls and accesses features of image processing, contains information about the state and properties of the camera, manages receiving raw frames from USB or SPI and sending them through image processing and then delivering them to the pipeline.

Lifecycle

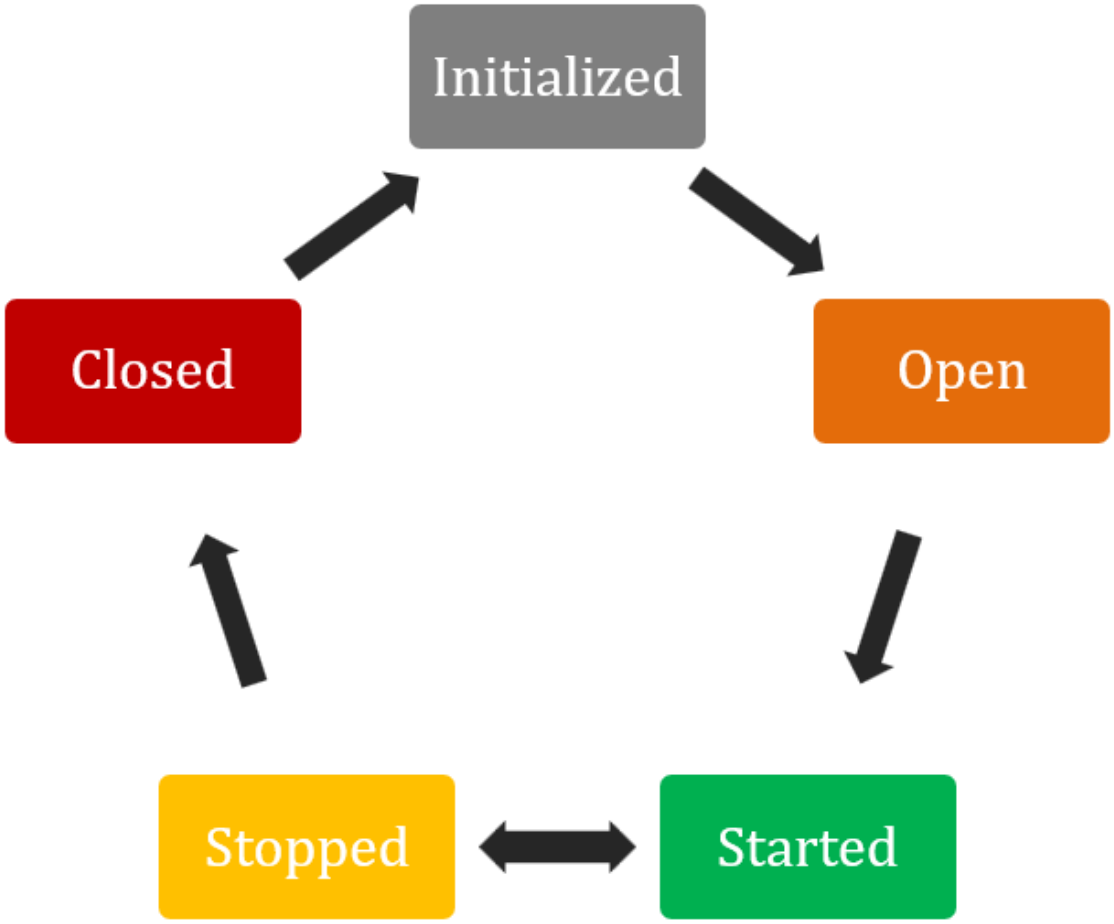


Figure 8.13 SeekCamera Lifecycle

The lifecycle management of a SeekCamera is handled automatically. The StateCallback allows for interfacing with different states of the camera:

- open() is called when the camera is created
- start() is called when the SeekCameraCaptureSession is created (and the application starts)
- stop() is called when the application stops
- close() is called when the camera is detached

8.16.2 Member Function Documentation

8.16.2.1 close()

```
synchronized void com.thermal.seekware.SeekCamera.close ( )
```

Closes the SeekCamera. Puts the camera to sleep (if supported) and frees all resources used by the camera and the imaging pipeline.

8.16.2.2 createSeekCameraCaptureSession() [1/2]

```
void com.thermal.seekware.SeekCamera.createSeekCameraCaptureSession (
            boolean getFiltered,
            boolean getTherm,
            boolean getColor,
            SeekPipelineListener...  listeners )
```

Creates a SeekCameraCaptureSession with the given parameters and starts the camera

Parameters

getFiltered	whether or not to request a 16bit, pre-AGC filtered image from the camera.
getTherm	whether or not to get the thermography image from the camera.
getColor	whether or not to get the color image from the camera.
listeners	the listeners to attach to the SeekCameraCaptureSession

8.16.2.3 createSeekCameraCaptureSession() [2/2]

```
void com.thermal.seekware.SeekCamera.createSeekCameraCaptureSession (
            SeekPipelineListener...  listeners )
```

Creates a SeekCameraCaptureSession with the given parameters and starts the camera

Parameters

listeners	the listeners to attach to the SeekCameraCaptureSession

8.16.2.4 getAspectRatio()

```
AspectRatio com.thermal.seekware.SeekCamera.getAspectRatio ( )
```

Gets the aspect ratio

Returns

the aspect ratio to display the image in

8.16.2.5 getCharacteristics()

```
Characteristics com.thermal.seekware.SeekCamera.getCharacteristics ( )
```

Gets the current characteristics

Returns

the characteristics associated with this SeekCamera

8.16.2.6 getChipId()

```
String com.thermal.seekware.SeekCamera.getChipId ( )
```

Gets the ChipId of the camera

Returns

String containing Chip ID from Sensor

8.16.2.7 getColorLut()

```
ColorLut com.thermal.seekware.SeekCamera.getColorLut ( )
```

Gets the current color lut used by the camera.

Returns

the color lut

8.16.2.8 getColorLutName()

```
String com.thermal.seekware.SeekCamera.getColorLutName ( )
```

Gets the current color lut used by the cameras name as a string

Returns

The current color lut's name as a String

8.16.2.9 getColorPalette() [1/2]

```
ByteBuffer com.thermal.seekware.SeekCamera.getColorPalette ( )
```

Gets a ByteBuffer of the current color LUT palette

Returns

a ByteBuffer with 256 colors representing the current color lut

8.16.2.10 getColorPalette() [2/2]

```
ByteBuffer com.thermal.seekware.SeekCamera.getColorPalette (
            ColorLut colorLut )
```

Gets a ByteBuffer of the desired color LUT palette

Returns

a ByteBuffer with 256 colors representing the given color lut

8.16.2.11 getCurrentState()

```
State com.thermal.seekware.SeekCamera.getCurrentState ( )
```

Gets the current camera state

Returns

the state the camera is currently in

8.16.2.12 getEmissivity()

```
float com.thermal.seekware.SeekCamera.getEmissivity ( )
```

Gets the current emissivity of the camera.

Returns

The emissivity

8.16.2.13 getImageSmoothing()

```
boolean com.thermal.seekware.SeekCamera.getImageSmoothing ( )
```

Gets the current image smoothing setting of the camera.

Returns

The image smoothing

8.16.2.14 loadUserLutData()

```
void com.thermal.seekware.SeekCamera.loadUserLutData (
            byte[] userLutData,
            ColorLut userLut )
```

Loads lut data into the given user lut

Parameters

userLutData	byte array that contains the user color lut
userLut	the desired lut to set the data for

See also

userLutData can be created for gradient ramps using ColorLut.createUserLut

8.16.2.15 memoryRead() [1/2]

```
synchronized void com.thermal.seekware.SeekCamera.memoryRead (
            MemoryRegion region,
            byte[] data ) throws UnsupportedOperationException
```

Reads data from the SeekCamera's internal memory at the specified region. NOTE: The function is a synchronous, blocking call and will only return when either the memory access has completed successfully or an error has occurred. The caller of this function is responsible to calling this function on a background thread, if desired.

Parameters

region	The memory region of the SeekCamera's internal memory to read from.
data	A buffer to store the data read from the SeekCamera.

Exceptions

UnsupportedOperationException	if reading from the specified region is not supported by the camera.

Exceptions

SeekIOException	if communication with the SeekCamera fails.
IllegalStateException	if the SeekCamera is Closed or Started. The internal memory of a SeekCamera can only be accessed in the Stopped state.

8.16.2.16 memoryRead() [2/2]

```
synchronized void com.thermal.seekware.SeekCamera.memoryRead (
            MemoryRegion region,
            int regionOffset,
            byte[] data ) throws UnsupportedOperationException
```

Reads data from the SeekCamera's internal memory at the specified region and offset. NOTE: The function is a synchronous, blocking call and will only return when either the memory access has completed successfully or an error has occurred. The caller of this function is responsible to calling this function on a background thread, if desired.

Parameters

region	The memory region of the SeekCamera's internal memory to read from.
regionOffset	Offset (in bytes) into the memory region to begin reading from.
data	A buffer to store the data read from the SeekCamera.

Exceptions

UnsupportedOperationException	if reading from the specified region is not supported by the current camera.
SeekIOException	if communication with the SeekCamera fails
IllegalStateException	if the SeekCamera is Closed or Started. The internal memory of a SeekCamera can only be accessed in the Stopped state.

8.16.2.17 memoryWrite() [1/2]

```
synchronized void com.thermal.seekware.SeekCamera.memoryWrite (
            MemoryRegion region,
            byte[] data )
```

Writes data to the SeekCamera's internal memory at the specified region. NOTE: The function is a synchronous, blocking call and will only return when either the memory access has completed successfully or an error has occurred. The caller of this function is responsible to calling this function on a background thread, if desired.

Parameters

region	The memory region of the SeekCamera's internal memory to write to.
data	The data to write to the SeekCamera's internal memory.

Exceptions

SeekIOException	if communication with the SeekCamera fails or the camera firmware does not support writing to the specified region
IllegalStateException	if the SeekCamera is Closed or Started. The internal memory of a SeekCamera can only be accessed in the Stopped state.

8.16.2.18 memoryWrite() [2/2]

```
synchronized void com.thermal.seekware.SeekCamera.memoryWrite (
            MemoryRegion region,
            int regionOffset,
            byte[] data )
```

Writes data to the SeekCamera's internal memory at the specified region and offset. NOTE: The function is a synchronous, blocking call and will only return when either the memory access has completed successfully or an error has occurred. The caller of this function is responsible to calling this function on a background thread, if desired.

Parameters

region	The memory region of the SeekCamera's internal memory to write to.
regionOffset	Offset (in bytes) into the memory region to begin writing to.
data	The data to write to the SeekCamera's internal memory.

Exceptions

SeekIOException	if communication with the SeekCamera fails or the camera firmware does not support writing to the specified region
IllegalStateException	if the SeekCamera is Closed or Started. The internal memory of a SeekCamera can only be accessed in the Stopped state.

8.16.2.19 open()

```
synchronized void com.thermal.seekware.SeekCamera.open ( )
```

Opens a SeekCamera and prepares it to be run.

Exceptions

InstantiationException	if the camera cannot be initialized
SeekIOException	if there is a problem communicating with the camera
IllegalStateException	if this SeekCamera is already open

8.16.2.20 reboot()

```
synchronized void com.thermal.seekware.SeekCamera.reboot (
            int msDelay )
```

Reboots a SeekCamera after the specified delay.

Parameters

msDelay	milliseconds to wait before rebooting the camera.

8.16.2.21 resumeShutter()

```
void com.thermal.seekware.SeekCamera.resumeShutter ( )
```

Resumes the shutter (USB cameras only)

8.16.2.22 setAGCMode()

```
void com.thermal.seekware.SeekCamera.setAGCMode (
            @NonNull AGCMode agcMode )
```

Sets the camera's AGC mode

Parameters

agcMode	the desired AGC mode

8.16.2.23 setAspectRatio()

```
void com.thermal.seekware.SeekCamera.setAspectRatio (
            @NonNull AspectRatio aspectRatio )
```

Sets the aspect ratio to the given value

Parameters

aspectRatio	the desired aspect ratio

8.16.2.24 setColorLut()

```
void com.thermal.seekware.SeekCamera.setColorLut (
            @NonNull ColorLut colorLut )
```

Sets the current color lut used by the camera.

Parameters

colorLut	the desired color lut

8.16.2.25 setEmissivity()

```
void com.thermal.seekware.SeekCamera.setEmissivity (
            float emissivity )
```

Sets the current emissivity of the camera.

Parameters

emissivity	the desired emissivity

8.16.2.26 setImageSmoothing()

```
void com.thermal.seekware.SeekCamera.setImageSmoothing (
            boolean imageSmoothing )
```

Sets the image smoothing setting of the camera.

Parameters

imageSmoothing	the desired image smoothing

8.16.2.27 setSeekExceptionListener()

```
void com.thermal.seekware.SeekCamera.setSeekExceptionListener (
            SeekExceptionListener seekExceptionListener )
```

Sets the SeekExceptionListener to the given parameter

Parameters

seekExceptionListener	the desired SeekExceptionListener

8.16.2.28 start()

```
synchronized void com.thermal.seekware.SeekCamera.start ( )
```

Starts the camera. Prepares the event driven imaging pipeline, wakes up the camera (if it is asleep), and begins processing image frames from the camera.

Exceptions

IllegalStateException	if this SeekCamera is not open

8.16.2.29 stop()

```
synchronized void com.thermal.seekware.SeekCamera.stop ( )
```

Stops the camera. Puts the cameras to sleep (if supported) and shuts down the imaging pipeline.

Exceptions

IllegalStateException	if this SeekCamera is not open

8.16.2.30 suspendShutter()

```
void com.thermal.seekware.SeekCamera.suspendShutter ( )
```

Suspends the shutter (USB cameras only)

8.16.2.31 toString()

```
String com.thermal.seekware.SeekCamera.toString ( )
```

Override default toString() method

Returns

a string with the model number, firmware version, and camera version

8.16.2.32 triggerShutter()

```
void com.thermal.seekware.SeekCamera.triggerShutter ( )
```

Triggers a shutter for a USB camera, or a flat field for a SPI camera

8.16.2.33 upgradeFirmware() [1/2]

```
void com.thermal.seekware.SeekCamera.upgradeFirmware (
            String firmwareFilename,
            boolean rebootAfterUpgrade )
```

Upgrades the firmware on this SeekCamera, then reboots the camera after 150ms, if requested

Parameters

firmwareFilename	the firmware file to upgrade with, found in assets
rebootAfterUpgrade	whether or not to reboot the camera after the upgrade is completed

8.16.2.34 upgradeFirmware() [2/2]

```
void com.thermal.seekware.SeekCamera.upgradeFirmware (
            String firmwareFilename,
            boolean rebootAfterUpgrade,
            int rebootMsDelay )
```

Upgrades the firmware on this SeekCamera, then reboots the camera after the given delay, if requested

Parameters

firmwareFilename	the firmware file to upgrade with, found in assets
rebootAfterUpgrade	whether or not to reboot the camera after the upgrade is completed
rebootMsDelay	the delay to wait in milliseconds for rebooting the camera (ignored if rebootAfterUpgrade is false)

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.17 com.thermal.seekware.SeekCameraManager Class Reference

Manages SeekCamera creation.

Inherits LifecycleObserver.

Public Member Functions

- SeekCameraManager (Context context, Handler handler, SeekCamera.StateCallback listener)
- int getUSBDeviceCount ()

Protected Attributes

- Context context

8.17.1 Detailed Description

Manages SeekCamera creation.

This class manages the SeekCameraReceiver that creates a SeekCamera when it is attached. Simply instantiate this class and it will automatically handle connecting to SPI and USB SeekCameras. Note: it is important that you store a reference to this object when created, even if there are no calls to it, as it will get garbage collected if you instantiate it in a local scope like onCreate.

The implementation of this class varies greatly from the camera2 android.hardware.camera2.CameraManager, as the USB cameras are not always connected, so it makes more sense to create them automatically in response to certain events (USB device attached, application start up (with a device already attached)).

8.17.2 Constructor & Destructor Documentation

8.17.2.1 SeekCameraManager()

```
com.thermal.seekware.SeekCameraManager.SeekCameraManager (
            Context context,
            Handler handler,
            SeekCamera.StateCallback listener )
```

Initializer Constructor

Parameters

context	application context
handler	application handler
listener	the listener to attach to the SeekCamera when it is created

8.17.3 Member Function Documentation

8.17.3.1 getUSBDeviceCount()

```
int com.thermal.seekware.SeekCameraManager.getUSBDeviceCount ( )
```

Gets the number of USB devices currently connected

Returns

the number of USB devices currently connected

8.17.4 Member Data Documentation

8.17.4.1 context

```
Context com.thermal.seekware.SeekCameraManager.context  [protected]
```

Application context, passed from the constructor

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCameraManager.java

8.18 com.thermal.seekware.SeekCamera.SeekExceptionListener Interface Reference

Listens for SeekIOExceptions.

Public Member Functions

- void onSeekIOException (SeekCamera camera, SeekIOException seekIOException)

8.18.1 Detailed Description

Listens for SeekIOExceptions.

8.18.2 Member Function Documentation

8.18.2.1 onSeekIOException()

```
void com.thermal.seekware.SeekCamera.SeekExceptionListener.onSeekIOException (
            SeekCamera camera,
            SeekIOException seekIOException )
```

Called when a SeekIOException occurs, on the same thread that the error occurred on

Parameters

camera	the SeekCamera associated with the callback
seekIOException	the SeekIOException generated

The documentation for this interface was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.19 com.thermal.seekware.SeekImage Class Reference

The object sent through the imaging pipeline.

Public Member Functions

- ByteBuffer getFilteredBuffer ()
- Thermography getThermography ()
- Bitmap getColorBitmap ()
- int getColorsUsed ()

8.19.1 Detailed Description

The object sent through the imaging pipeline.

This class is what is sent through the imaging pipeline, originating from a SeekCameraCaptureSession and going through as many SeekPipelines as specified and then to an output such as SeekImageView. It contains all information related to the thermal image.

8.19.2 Member Function Documentation

8.19.2.1 getColorBitmap()

```
Bitmap com.thermal.seekware.SeekImage.getColorBitmap ( )
```

Gets the color bitmap and returns it

Returns

the color bitmap of this image

8.19.2.2 getColorsUsed()

```
int com.thermal.seekware.SeekImage.getColorsUsed ( )
```

Gets the number of colors used for the color bar and returns it

Returns

the number of colors used for the color bar (0-256)

8.19.2.3 getFilteredBuffer()

```
ByteBuffer com.thermal.seekware.SeekImage.getFilteredBuffer ( )
```

Gets the filtered buffer and returns it Note: this is formatted like a flat byte array, so use SeekUtility.flatten to convert a 2D Point into a flat index.

Returns

the filtered buffer associated with this image

8.19.2.4 getThermography()

```
Thermography com.thermal.seekware.SeekImage.getThermography ( )
```

Gets the thermography and returns it. Note: The thermography will only be valid for this image only. If you plan on using values from this class for longer than one frame time, you must make a copy or there will be a chance that the data will be overridden.

Returns

the thermography associated with this image

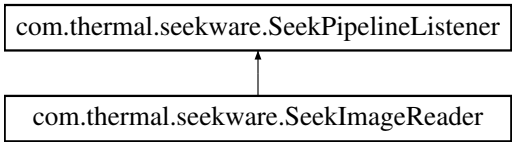
The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekImage.java

8.20 com.thermal.seekware.SeekImageReader Class Reference

Provides imaging and thermogaphy from the camera directly to the user.

Inheritance diagram for com.thermal.seekware.SeekImageReader:



Classes

- interface OnImageAvailableListener

  Called when a SeekImage is available.

Public Member Functions

- SeekImageReader ()
- SeekImageReader (SeekPipeline input)
- void setOnImageAvailableListener (OnImageAvailableListener onImageAvailableListener)
- void onImageSent (SeekImage image)

8.20.1 Detailed Description

Provides imaging and thermogaphy from the camera directly to the user.

This class's primary function is to provide direct access to thermal data to the end user. NOTE: You must specifically request for the filtered buffer in SeekCameraCaptureSession or else the filtered buffer in the SeekImage provided by the callback will be null.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 SeekImageReader() [1/2]

```
com.thermal.seekware.SeekImageReader.SeekImageReader ( )
```

Default Constructor

8.20.2.2 SeekImageReader() [2/2]

```
com.thermal.seekware.SeekImageReader.SeekImageReader (
            SeekPipeline input )
```

Pipeline Constructor

Parameters

input	where to listen for input

8.20.3 Member Function Documentation

8.20.3.1 onImageSent()

```
void com.thermal.seekware.SeekImageReader.onImageSent (
            SeekImage image )
```

Callback for seekPipeline.sendImage()

Parameters

image	the SeekImage associated with the callback

Implements com.thermal.seekware.SeekPipelineListener.

8.20.3.2 setOnImageAvailableListener()

```
void com.thermal.seekware.SeekImageReader.setOnImageAvailableListener (
            OnImageAvailableListener onImageAvailableListener )
```

Sets the OnImageAvailableListener

Parameters

onImageAvailableListener	the desired listener for this class

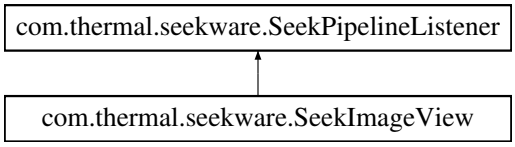
The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekImageReader.java

8.21 com.thermal.seekware.SeekImageView Class Reference

Draws a thermal image to the screen.

Inheritance diagram for com.thermal.seekware.SeekImageView:



Classes

- interface OnFrameAvailableListener

  Listens for an available frame.

Public Member Functions

- SeekImageView (Context context, SeekPipeline input)
- SeekImageView (Context context)
- SeekImageView (Context context, AttributeSet attrs)
- void setInput (SeekPipeline input)
- void onImageSent (SeekImage image)
- void setOnFrameAvailableListener (OnFrameAvailableListener onFrameAvailableListener)

8.21.1 Detailed Description

Draws a thermal image to the screen.

This class processes SeekImages from either a SeekCameraCaptureSession or a SeekPipeline and renders them into a GLSurfaceView.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 SeekImageView() [1/3]

```
com.thermal.seekware.SeekImageView.SeekImageView (
            Context context,
            SeekPipeline input )
```

SeekPipeline Constructor

Parameters

context	application context
input	imaging pipeline object to render

8.21.2.2 SeekImageView() [2/3]

```
com.thermal.seekware.SeekImageView.SeekImageView (
            Context context )
```

SeekCameraCaptureSession Constructor

Parameters

context	application context

8.21.2.3 SeekImageView() [3/3]

```
com.thermal.seekware.SeekImageView.SeekImageView (
            Context context,
            AttributeSet attrs )
```

Default View Constructor Called when a view is added to an xml layout.

Parameters

context	application context
attrs	attribute set

8.21.3 Member Function Documentation

8.21.3.1 onImageSent()

```
void com.thermal.seekware.SeekImageView.onImageSent (
            SeekImage image )
```

Callback for seekPipeline.sendImage()

Parameters

image	the SeekImage associated with the callback

Implements com.thermal.seekware.SeekPipelineListener.

8.21.3.2 setInput()

```
void com.thermal.seekware.SeekImageView.setInput (
            SeekPipeline input )
```

Updates the rendering input for this View

Parameters

input	imaging pipeline object to receive data from

8.21.3.3 setOnFrameAvailableListener()

```
void com.thermal.seekware.SeekImageView.setOnFrameAvailableListener (
            OnFrameAvailableListener onFrameAvailableListener )
```

Sets the OnFrameAvailableListener to the given parameter

Parameters

onFrameAvailableListener	the desired listener

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekImageView.java

8.22 com.thermal.seekware.SeekIOException Class Reference

IOException with custom error codes.

Inherits IOException.

Classes

- enum ErrorCode

  Error codes that describe communication failures between the SDK and a SeekCamera.

8.22.1 Detailed Description

IOException with custom error codes.

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekIOException.java

8.23 com.thermal.seekware.SeekLogger Class Reference

Controls and filters logging.

Static Public Member Functions

- static void setFilter (int filter)
- static void addTagFilters (String... tags)
- static void verbose (String tag, String msg)
- static void debug (String tag, String msg)
- static void info (String tag, String msg)
- static void warn (String tag, String msg)
- static void error (String tag, String msg)

Static Public Attributes

- static final int ALL = 0x11111
- static final int NONE = 0x00000

8.23.1 Detailed Description

Controls and filters logging.

This class is a wrapper around the Android Log that allows for filtering by channel as well as filtering by tag.

8.23.2 Member Function Documentation

8.23.2.1 addTagFilters()

```
static void com.thermal.seekware.SeekLogger.addTagFilters (
            String...  tags )  [static]
```

Adds a filter that only logs the messages that match the desired tags.

Parameters

tags	tags that should be logged

8.23.2.2 debug()

```
static void com.thermal.seekware.SeekLogger.debug (
            String tag,
            String msg )  [static]
```

Logs to the debug channel if the filters allow

Parameters

tag	class tag
msg	message to log

8.23.2.3 error()

```
static void com.thermal.seekware.SeekLogger.error (
```

```
            String tag,
            String msg )  [static]
```

Logs to the error channel if the filters allow

Parameters

tag	class tag
msg	message to log

8.23.2.4 info()

```
static void com.thermal.seekware.SeekLogger.info (
            String tag,
            String msg )  [static]
```

Logs to the info channel if the filters allow

Parameters

tag	class tag
msg	message to log

8.23.2.5 setFilter()

```
static void com.thermal.seekware.SeekLogger.setFilter (
            int filter )  [static]
```

Sets the filter for logging specific channels (debug, info, warning, error, verbose). To filter for only certain channels, you can bitwise or the constants in SeekLogger to get only the specified channels. For example, if you only want debug and info logs, you would change the following line to: SeekLogger.setFilter(SeekLogger.DEBUG | Seek↩ Logger.INFO);

Parameters

filter	the filter to apply to all logging

8.23.2.6 verbose()

```
static void com.thermal.seekware.SeekLogger.verbose (
            String tag,
            String msg )  [static]
```

Logs to the verbose channel if the filters allow

Parameters

tag	class tag
msg	message to log

8.23.2.7 warn()

```
static void com.thermal.seekware.SeekLogger.warn (
            String tag,
            String msg )  [static]
```

Logs to the warning channel if the filters allow

Parameters

tag	class tag
msg	message to log

8.23.3 Member Data Documentation

8.23.3.1 ALL

```
final int com.thermal.seekware.SeekLogger.ALL = 0x11111  [static]
```

Turns on all logging

8.23.3.2 NONE

```
final int com.thermal.seekware.SeekLogger.NONE = 0x00000  [static]
```

Turns off all logging

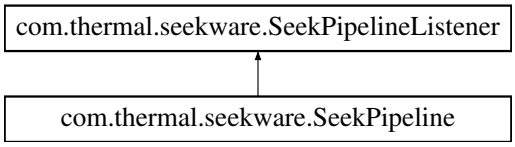
The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekLogger.java

8.24 com.thermal.seekware.SeekPipeline Class Reference

Extend this class to do your own processing on a SeekCamera by overriding process().

Inheritance diagram for com.thermal.seekware.SeekPipeline:



Public Member Functions

- void onImageSent (SeekImage image)

Protected Attributes

- SeekPipeline input
- SeekImage image

8.24.1 Detailed Description

Extend this class to do your own processing on a SeekCamera by overriding process().

This abstract class is the building block for processing SeekImages in the SDK imaging pipeline. If you would like to design and insert your own processing steps that operate on the thermal or imaging data from a SeekCamera, you can extend this class and insert your own processing steps.

NOTE: Any processing steps you design must be less than 1 frame time. Timing Requirements on process(): $<$ 8fps: 120ms $<$ 9fps: 110ms $<$ 16fps: 60ms $<$ 18fps: 50ms $<$ 27fps: 35ms

For compatibility with all SeekCamera's we recommend no more that 30ms of processing.

8.24.2 Member Function Documentation

8.24.2.1 onImageSent()

```
void com.thermal.seekware.SeekPipeline.onImageSent (
            SeekImage image )
```

Callback for seekPipeline.sendImage()

Parameters

image	the SeekImage associated with the callback

Implements com.thermal.seekware.SeekPipelineListener.

8.24.3 Member Data Documentation

8.24.3.1 image

```
SeekImage com.thermal.seekware.SeekPipeline.image  [protected]
```

The SeekImage that will be sent after this has finished processing

8.24.3.2 input

```
SeekPipeline com.thermal.seekware.SeekPipeline.input  [protected]
```

Where the input should be received from. If this is null, this SeekPipeline should be directly attached to a SeekCameraCaptureSession

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekPipeline.java

8.25 com.thermal.seekware.SeekPipelineException Class Reference

Runtime exception with custom error codes.

Inherits RuntimeException.

Classes

- enum ErrorCode

  Custom error codes.

Public Member Functions

- ErrorCode getErrorCode ()

8.25.1 Detailed Description

Runtime exception with custom error codes.

8.25.2 Member Function Documentation

8.25.2.1 getErrorCode()

```
ErrorCode com.thermal.seekware.SeekPipelineException.getErrorCode ( )
```

Gets the error code associated with this exception

Returns

the error code associated with this exception

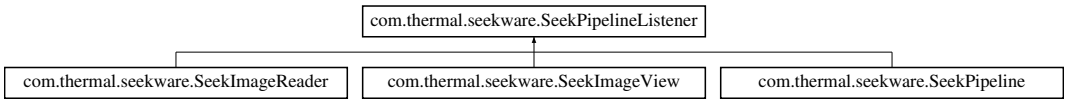
The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekPipelineException.java

8.26 com.thermal.seekware.SeekPipelineListener Interface Reference

Allows for callbacks between SeekPipeline objects when a SeekImage is ready for processing.

Inheritance diagram for com.thermal.seekware.SeekPipelineListener:



Public Member Functions

- void onImageSent (SeekImage image)

8.26.1 Detailed Description

Allows for callbacks between SeekPipeline objects when a SeekImage is ready for processing.

8.26.2 Member Function Documentation

8.26.2.1 onImageSent()

```
void com.thermal.seekware.SeekPipelineListener.onImageSent (
            SeekImage image )
```

Callback for seekPipeline.sendImage()

Parameters

image	the SeekImage associated with the callback

Implemented in com.thermal.seekware.SeekPipeline, com.thermal.seekware.SeekImageView, and com.thermal.seekware.SeekImag

The documentation for this interface was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekPipelineListener.java

8.27 com.thermal.seekware.SeekUtility Class Reference

Contains utility classes, functions, for use with this SDK.

Classes

- class OrientationManager

  Manages orientation of the devices and rotation of views and overlays.
- class PermissionHandler

  Handles Android 6+ permissions.
- class Temperature

  Represents an immutable temperature with a value and a unit.

Static Public Member Functions

- static void saveByteArray (String filename, byte[ ] data)
- static void saveByteBufferAsPrivateRaw (@NonNull ByteBuffer data, @NonNull String filename, @NonNull Context context)
- static String generateFilename ()
- static Bitmap bitmapFromUri (@NonNull Uri uri, @NonNull Context context)
- static Bitmap getBitmapFromDrawable (Context context, int drawableId) throws IllegalArgumentException
- static Bitmap overlayBitmapToCenter (@NonNull Bitmap bitmap1, @NonNull Bitmap bitmap2)
- static Bitmap createVideoThumbnail (@NonNull Uri uri, @NonNull Context context, @NonNull Size size)
- static Bitmap resizeBitmap (@NonNull Bitmap image, int maxWidth, int maxHeight)
- static Bitmap createSquaredBitmap (@NonNull Bitmap source)
- static Bitmap rotateBitmap (@NonNull Bitmap source, float angle)
- static Bitmap flipBitmapHorizontal (@NonNull Bitmap source)
- static Bitmap flipBitmapVertical (@NonNull Bitmap source)
- static ArrayList< String > findFirmwareUpgradeFiles (Context context, String type)
- static String enumNameToString (String enumName)
- static float distance (Point p1, Point p2)
- static int closest (float val, int... candidates)
- static ByteBuffer allocateByteBuffer (int bytes)
- static byte[ ] byteArrayFromFile (String filename, Context context) throws IOException
- static ByteBuffer byteBufferFromFile (String filename, Context context) throws IOException
- static int getDeviceDefaultOrientation (Context context)

8.27.1 Detailed Description

Contains utility classes, functions, for use with this SDK.

8.27.2 Member Function Documentation

8.27.2.1 allocateByteBuffer()

```
static ByteBuffer com.thermal.seekware.SeekUtility.allocateByteBuffer (
            int bytes )  [static]
```

Allocates a ByteBuffer suitable for use with SeekCameras

Parameters

bytes	number of bytes to allocate

Returns

an allocated ByteBuffer in little endian order

8.27.2.2 bitmapFromUri()

```
static Bitmap com.thermal.seekware.SeekUtility.bitmapFromUri (
            @NonNull Uri uri,
            @NonNull Context context )  [static]
```

Creates a bitmap from the given uri or returns null upon failure

Parameters

uri	desired uri
context	application context

Returns

bitmap from the given uri

8.27.2.3 byteArrayFromFile()

```
static byte [] com.thermal.seekware.SeekUtility.byteArrayFromFile (
            String filename,
            Context context ) throws IOException  [static]
```

Reads a .raw file into a byte array and returns it.

Parameters

filename	the file to read in assets
context	application context
size	size of the ByteBuffer

Returns

a ByteBuffer with the data from the file

8.27.2.4 byteBufferFromFile()

```
static ByteBuffer com.thermal.seekware.SeekUtility.byteBufferFromFile (
            String filename,
            Context context ) throws IOException  [static]
```

Reads a .raw file into a ByteBuffer and returns it.

Parameters

filename	the file to read
context	application context
size	size of the ByteBuffer

Returns

a ByteBuffer with the data from the file

8.27.2.5 closest()

```
static int com.thermal.seekware.SeekUtility.closest (
            float val,
            int...  candidates )  [static]
```

Returns the closest candidate to the given value

Parameters

val	value to compare against candidates
candidates	any number of candidates to check

Returns

the closest candidate to the given value

8.27.2.6 createSquaredBitmap()

```
static Bitmap com.thermal.seekware.SeekUtility.createSquaredBitmap (
            @NonNull Bitmap source )  [static]
```

Creates a square bitmap from the source bitmap.

Returns

a square bitmap

8.27.2.7 createVideoThumbnail()

```
static Bitmap com.thermal.seekware.SeekUtility.createVideoThumbnail (
            @NonNull Uri uri,
            @NonNull Context context,
            @NonNull Size size )  [static]
```

Creates a video thumbnail with a play icon overlay.

Parameters

uri	video uri to generate thumbnail for
context	application context
size	size of the thumbnail in pixels

Returns

a bitmap of the video thumbnail

8.27.2.8 distance()

```
static float com.thermal.seekware.SeekUtility.distance (
            Point p1,
            Point p2 )  [static]
```

Calculates the distance between two points and returns it

Parameters

p1	first point
p2	second point

Returns

the distance between the two points

8.27.2.9 enumNameToString()

```
static String com.thermal.seekware.SeekUtility.enumNameToString (
            String enumName )  [static]
```

Takes an enum name, converts it to a more readable format, and returns it For example, enum COLOR_LUT will return "Color lut"

Parameters

enumName	enum.name() of the enum to convert

Returns

String representing a cleaned up version of the enum name

8.27.2.10 findFirmwareUpgradeFiles()

```
static ArrayList<String> com.thermal.seekware.SeekUtility.findFirmwareUpgradeFiles (
            Context context,
            String type )  [static]
```

Generates a list of firmware upgrade files found in assets

Parameters

context	application context
type	type of file to search for (start of name)

Returns

a list of firmware upgrade files

8.27.2.11 flipBitmapHorizontal()

```
static Bitmap com.thermal.seekware.SeekUtility.flipBitmapHorizontal (
            @NonNull Bitmap source )  [static]
```

Flips the source bitmap horizontally.

Returns

the source bitmap flipped horizontally

8.27.2.12 flipBitmapVertical()

```
static Bitmap com.thermal.seekware.SeekUtility.flipBitmapVertical (
            @NonNull Bitmap source )  [static]
```

Flips the source bitmap vertically.

Returns

the source bitmap flipped vertically

8.27.2.13 generateFilename()

```
static String com.thermal.seekware.SeekUtility.generateFilename ( )  [static]
```

Generates a default file name Format: IR_yyyyMMdd_HHmmss

Returns

the generated file name

8.27.2.14 getBitmapFromDrawable()

```
static Bitmap com.thermal.seekware.SeekUtility.getBitmapFromDrawable (
            Context context,
            int drawableId ) throws IllegalArgumentException  [static]
```

Finds a drawable in the application resources and converts it to a bitmap.

Parameters

context	application context
drawableId	id of the drawable to convert

Returns

a bitmap of the drawable

8.27.2.15 getDeviceDefaultOrientation()

```
static int com.thermal.seekware.SeekUtility.getDeviceDefaultOrientation (
            Context context )  [static]
```

Gets the device's default orientation and returns it

Parameters

context	application context

Returns

the device's default orientation

8.27.2.16 overlayBitmapToCenter()

```
static Bitmap com.thermal.seekware.SeekUtility.overlayBitmapToCenter (
            @NonNull Bitmap bitmap1,
            @NonNull Bitmap bitmap2 )  [static]
```

Overlays bitmap2 into bitmap1.

Parameters

bitmap1	bitmap to draw into
bitmap2	bitmap to draw

Returns

a new bitmap with bitmap2 overlaid into bitmap1

8.27.2.17 resizeBitmap()

```
static Bitmap com.thermal.seekware.SeekUtility.resizeBitmap (
            @NonNull Bitmap image,
            int maxWidth,
            int maxHeight )  [static]
```

Resizes a bitmap to the given width and height.

Parameters

image	desired bitmap to resize
maxWidth	max width to rescale to
maxHeight	max height to rescale to

Returns

scaled bitmap

8.27.2.18 rotateBitmap()

```
static Bitmap com.thermal.seekware.SeekUtility.rotateBitmap (
            @NonNull Bitmap source,
            float angle )  [static]
```

Rotates the source bitmap by the given angle.

Returns

the source bitmap rotated by the given angle

8.27.2.19 saveByteArray()

```
static void com.thermal.seekware.SeekUtility.saveByteArray (
            String filename,
            byte[] data )  [static]
```

Writes the given binary frame to the given file name

Parameters

filename	the desired filename
data	the desired frame

8.27.2.20 saveByteBufferAsPrivateRaw()

```
static void com.thermal.seekware.SeekUtility.saveByteBufferAsPrivateRaw (
            @NonNull ByteBuffer data,
            @NonNull String filename,
            @NonNull Context context )  [static]
```

Writes the given binary frame to the given file name

Parameters

data	the desired frame
filename	the desired filename
context	application context

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekUtility.java

8.28 com.thermal.seekware.SeekCamera.State Enum Reference

The current state of the camera.

Public Member Functions

- boolean isAtLeast (@NonNull State state)

Public Attributes

- INITIALIZED
- OPENED
- STARTED
- STOPPED
- CLOSED

8.28.1 Detailed Description

The current state of the camera.

In general, the basic lifecycle of a SeekCamera looks like this: Initialized -> Open -> Started <--> Stopped -> Closed

8.28.2 Member Function Documentation

8.28.2.1 isAtLeast()

```
boolean com.thermal.seekware.SeekCamera.State.isAtLeast (
            @NonNull State state )
```

< Closed Compares the current state to given state and returns whether or not the current state is at least the given one Meant to mirror the call to LifecycleOwner's isAtLeast function

Parameters

state	state to compare to

Returns

whether or not the state is at least the given state

8.28.3 Member Data Documentation

8.28.3.1 CLOSED

`com.thermal.seekware.SeekCamera.State.CLOSED`

Set at the end of close()

8.28.3.2 INITIALIZED

`com.thermal.seekware.SeekCamera.State.INITIALIZED`

Set at the end of the constructor Initialized

8.28.3.3 OPENED

`com.thermal.seekware.SeekCamera.State.OPENED`

Set at the end of open() Opened

8.28.3.4 STARTED

`com.thermal.seekware.SeekCamera.State.STARTED`

Set at the end of start() Started

8.28.3.5 STOPPED

`com.thermal.seekware.SeekCamera.State.STOPPED`

Set at the end of stop() Stopped

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.29 com.thermal.seekware.SeekUtility.PermissionHandler.StateCallback Interface Reference

Callback for when a permission is granted.

8.29.1 Detailed Description

Callback for when a permission is granted.

The documentation for this interface was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekUtility.java

8.30 com.thermal.seekware.SeekCamera.StateCallback Interface Reference

Allows for callbacks to the states of the camera.

Public Member Functions

- void onInitialized (SeekCamera camera)
- void onOpened (SeekCamera camera)
- void onStarted (SeekCamera camera)
- void onStopped (SeekCamera camera)
- void onClosed (SeekCamera camera)
- void onMemoryAccess (SeekCamera camera, MemoryRegion region, int progress)
- void onReboot (SeekCamera seekCamera)
- void onError (SeekCamera camera, Exception e)

8.30.1 Detailed Description

Allows for callbacks to the states of the camera.

This interface allows for callbacks to certain events of a SeekCamera. Meant to mimic the interface Camera↩ Device.StateCallback from the camera2 API. NOTE: You must add the listener with the function addListener(↩ StateCallback stateCallback) in order for the callbacks to work properly.

8.30.2 Member Function Documentation

8.30.2.1 onClosed()

```
void com.thermal.seekware.SeekCamera.StateCallback.onClosed (
            SeekCamera camera )
```

Called after the camera is closed.

Parameters

camera	the SeekCamera associated with the callback

8.30.2.2 onError()

```
void com.thermal.seekware.SeekCamera.StateCallback.onError (
            SeekCamera camera,
            Exception e )
```

Called when an error occurs involving the state of the camera. Exceptions that are generated will not be thrown, but instead passed to this callback. Functions that call onError will also return, as if an error has been thrown and caught.

Parameters

camera	the SeekCamera associated with the callback
e	Exception thrown

8.30.2.3 onInitialized()

```
void com.thermal.seekware.SeekCamera.StateCallback.onInitialized (
            SeekCamera camera )
```

Called after the camera is initialized.

Parameters

camera	the SeekCamera associated with the callback

8.30.2.4 onMemoryAccess()

```
void com.thermal.seekware.SeekCamera.StateCallback.onMemoryAccess (
            SeekCamera camera,
            MemoryRegion region,
            int progress )
```

Sends progress updates during a firmware memory read/write.

Parameters

camera	the SeekCamera associated with the callback
region	the MemoryRegion that was accessed
progress	the percent completion of the memory read/write

8.30.2.5 onOpened()

```
void com.thermal.seekware.SeekCamera.StateCallback.onOpened (
            SeekCamera camera )
```

Called after the camera is opened.

Parameters

camera	the SeekCamera associated with the callback

8.30.2.6 onReboot()

```
void com.thermal.seekware.SeekCamera.StateCallback.onReboot (
            SeekCamera seekCamera )
```

Called when a reboot of the camera is requested

Parameters

seekCamera	the SeekCamera associated with the callback

8.30.2.7 onStarted()

```
void com.thermal.seekware.SeekCamera.StateCallback.onStarted (
            SeekCamera camera )
```

Called after the camera is started.

Parameters

camera	the SeekCamera associated with the callback

8.30.2.8 onStopped()

```
void com.thermal.seekware.SeekCamera.StateCallback.onStopped (
            SeekCamera camera )
```

Called after the camera is stopped.

Parameters

camera	the SeekCamera associated with the callback

The documentation for this interface was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekCamera.java

8.31 com.thermal.seekware.SeekUtility.Temperature Class Reference

Represents an immutable temperature with a value and a unit.

Classes

- enum Unit

  Represents a temperature unit (C, F or K)

Public Member Functions

- Temperature (float value, Unit unit)
- Temperature (float value)
- void setUnit (Unit unit)
- Unit getUnit ()
- float getValue ()

8.31.1 Detailed Description

Represents an immutable temperature with a value and a unit.

This class represents a temperature with a value and a unit and allows for automatic conversion between different units with the convertTemperature(unit) function.

8.31.2 Constructor & Destructor Documentation

8.31.2.1 Temperature() [1/2]

```
com.thermal.seekware.SeekUtility.Temperature.Temperature (
            float value,
            Unit unit )
```

Initializer Constructor

Parameters

value	the initial temperature value
unit	the initial temperature unit

8.31.2.2 Temperature() [2/2]

```
com.thermal.seekware.SeekUtility.Temperature.Temperature (
            float value )
```

Partial Initializer Constructor

Parameters

value	the initial temperature value in degrees Celsius.

8.31.3 Member Function Documentation

8.31.3.1 getUnit()

```
Unit com.thermal.seekware.SeekUtility.Temperature.getUnit ( )
```

Gets the current units used to compute a temperature value.

Returns

the temperature unit

8.31.3.2 getValue()

```
float com.thermal.seekware.SeekUtility.Temperature.getValue ( )
```

Gets the current temperature value based on the current units.

Returns

the temperature unit

8.31.3.3 setUnit()

```
void com.thermal.seekware.SeekUtility.Temperature.setUnit (
            Unit unit )
```

Sets the units used to compute value.

Parameters

unit	the desired temperature units

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekUtility.java

8.32 com.thermal.seekware.Thermography Class Reference

Holds thermography data from a SeekCamera.

Classes

- class Metrics

  Simple struct to hold average, min, and max temperatures.

Public Member Functions

- ByteBuffer getBuffer ()
- SeekUtility.Temperature getSpotTemp ()
- SeekUtility.Temperature getMinTemp ()
- SeekUtility.Temperature getMaxTemp ()
- Point getMinPoint ()
- Point getMaxPoint ()
- SeekUtility.Temperature calcSpotTemperature (Point point, int diameter)
- Metrics calcSpotMetrics (Point point, int diameter)
- SeekUtility.Temperature calcAreaTemperature (Point point, int width, int height)
- Metrics calcAreaMetrics (Point point, int width, int height)
- int calculateIndex (int x, int y)
- SeekUtility.Temperature getPointTemperature (Point point)

Static Public Member Functions

- static float fromShort (short shortTemp)
- static Thermography fromFile (@NonNull String bufferFilename, @NonNull ExifInterface exifInterface, @NonNull Context context)

Static Public Attributes

- static final int THERMOGRAPHY_SCALE = 64
- static final int THERMOGRAPHY_OFFSET = 40

8.32.1 Detailed Description

Holds thermography data from a SeekCamera.

8.32.2 Member Function Documentation

8.32.2.1 calcAreaMetrics()

```
Metrics com.thermal.seekware.Thermography.calcAreaMetrics (
            Point point,
            int width,
            int height )
```

Calculates the average, min, and max temperature of the given rectangle.

Parameters

point	(x,y) coordinates of the center of the rectangle
width	width of the rectangle
height	height of the rectangle

Returns

the metrics of the given rectangle

8.32.2.2 calcAreaTemperature()

```
SeekUtility.Temperature com.thermal.seekware.Thermography.calcAreaTemperature (
            Point point,
            int width,
            int height )
```

Calculates the area temperature of the given rectangle. The same calculation as Thermography#calcSpotTemperature(Point, int), but allows for any rectangle rather than only squares.

Parameters

point	x,y) coordinates of the center of the area
width	width of the rectangle to average
height	height of the rectangle to average

Returns

average temperature of the rectangle

8.32.2.3 calcSpotMetrics()

```
Metrics com.thermal.seekware.Thermography.calcSpotMetrics (
            Point point,
            int diameter )
```

Equivalent to Thermography#calcAreaMetrics(Point, int, int) with width and height as diameter.

Parameters

point	(x,y) coordinates of the center of the spot
diameter	diameter of the square

Returns

average, min, and max temperature of the diameter grid of pixels

8.32.2.4 calcSpotTemperature()

```
SeekUtility.Temperature com.thermal.seekware.Thermography.calcSpotTemperature (
            Point point,
            int diameter )
```

Calculates the spot temperature based on the given parameters

Pixel Average Diagram

Table 8.90 1 x 1 Pixel Average

	x-2	x-1	x	x+1	x+2	x+3
y-2						
y-1						
y			*			
y+1						
y+2						
y+3						

Table 8.91 2 x 2 Pixel Average

	x-2	x-1	x	x+1	x+2	x+3
y-2						
y-1						
y			*	*		
y+1			*	*		
y+2						
y+3						

Table 8.92 3 x 3 Pixel Average

	x-2	x-1	x	x+1	x+2	x+3
y-2						
y-1		*	*	*		
y		*	*	*		
y+1		*	*	*		
y+2						
y+3						

Table 8.93 4 x 4 Pixel Average

	x-2	x-1	x	x+1	x+2	x+3
y-2						
y-1		*	*	*	*	
y		*	*	*	*	
y+1		*	*	*	*	
y+2		*	*	*	*	
y+3						

Parameters

point	(x,y) coordinates of the center of the spot
diameter	diameter to average

Returns

average temperature of the diameter grid of pixels

8.32.2.5 calculateIndex()

```
int com.thermal.seekware.Thermography.calculateIndex (
            int x,
            int y )
```

Calculates a flattened index from x, y (column, row) coordinates

Parameters

x	horizontal (column) coordinate
y	vertical (row) coordinate

Returns

integer index value

8.32.2.6 fromFile()

```
static Thermography com.thermal.seekware.Thermography.fromFile (
            @NonNull String bufferFilename,
            @NonNull ExifInterface exifInterface,
            @NonNull Context context )  [static]
```

Creates a thermography object based on the given files

Parameters

bufferFilename	the file of the thermography buffer (.raw file)
exifInterface	Exif tags with thermography data
context	application context

Returns

Thermography with parameters generated from the files

8.32.2.7 fromShort()

```
static float com.thermal.seekware.Thermography.fromShort (
            short shortTemp )  [static]
```

Converts a short temp into a float value

Parameters

shortTemp	short temp

Returns

float value

8.32.2.8 getBuffer()

```
ByteBuffer com.thermal.seekware.Thermography.getBuffer ( )
```

Returns a ByteBuffer of the thermography values. In order to get a floating point value out of this buffer at (x, y), the following steps need to be taken:

1. Calculate the corrected point by using SeekCamera.Characteristics#getOrientation().
2. convert the ByteBuffer returned here to a short buffer (getBuffer().asShortBuffer()).

3. flatten the 2D Point using calculateIndex(int, int).
4. Get the short value out of the short buffer (getBuffer().asShortBuffer().get(index)).
5. Convert it to a float with fromShort(short).
6. The float value is now ready to use.

Returns

a ByteBuffer of the thermography values

8.32.2.9 getMaxPoint()

```
Point com.thermal.seekware.Thermography.getMaxPoint ( )
```

Returns a Point of where the hottest coordinate is located.

Returns

a Point of where the hottest coordinate is located

8.32.2.10 getMaxTemp()

```
SeekUtility.Temperature com.thermal.seekware.Thermography.getMaxTemp ( )
```

Gets the 5x5 average temperature around the hottest pixel coordinate in the image. This is equivalent to calling calcSpotTemperature(Point, int) with getMaxPoint() and 5.

Returns

the 5x5 average temperature around the hottest pixel coordinate

8.32.2.11 getMinPoint()

```
Point com.thermal.seekware.Thermography.getMinPoint ( )
```

Returns a Point of where the coldest coordinate is located.

Returns

a Point of where the coldest coordinate is located

8.32.2.12 getMinTemp()

```
SeekUtility.Temperature com.thermal.seekware.Thermography.getMinTemp ( )
```

Gets the 5x5 average temperature around the coldest pixel coordinate in the image. This is equivalent calling calcSpotTemperature(Point, int) with getMinPoint() and 5.

Returns

the 5x5 average temperature around the coldest pixel coordinate

8.32.2.13 getPointTemperature()

```
SeekUtility.Temperature com.thermal.seekware.Thermography.getPointTemperature (
            Point point )
```

Gets the point temperature based on the given parameters

Parameters

point	(x,y) coordinates of the center of the spot

Returns

temperature of the given point

8.32.2.14 getSpotTemp()

```
SeekUtility.Temperature com.thermal.seekware.Thermography.getSpotTemp ( )
```

Gets the 6x6 average temperature around the center pixel coordinate and returns it. This is equivalent to calling calcSpotTemperature(Point, int) with new Point(SeekCamera.Characteristics#getWidth() / 2, SeekCamera.Characteristics#getHeight() / 2) and 6.

Returns

the 6x6 average temperature around the center pixel coordinate

8.32.3 Member Data Documentation

8.32.3.1 THERMOGRAPHY_OFFSET

```
final int com.thermal.seekware.Thermography.THERMOGRAPHY_OFFSET = 40  [static]
```

The thermography offset used when converting to a float

8.32.3.2 THERMOGRAPHY_SCALE

```
final int com.thermal.seekware.Thermography.THERMOGRAPHY_SCALE = 64  [static]
```

The thermography scale used when converting to a float

The documentation for this class was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/Thermography.java

8.33 com.thermal.seekware.SeekUtility.Temperature.Unit Enum Reference

Represents a temperature unit (C, F or K)

Public Attributes

- CELSIUS
- FAHRENHEIT
- KELVIN

8.33.1 Detailed Description

Represents a temperature unit (C, F or K)

This enum represents the units of a Temperature.

8.33.2 Member Data Documentation

8.33.2.1 CELSIUS

```
com.thermal.seekware.SeekUtility.Temperature.Unit.CELSIUS
```

Celsius (C)

8.33.2.2 FAHRENHEIT

```
com.thermal.seekware.SeekUtility.Temperature.Unit.FAHRENHEIT
```

Fahrenheit (F)

8.33.2.3 KELVIN

```
com.thermal.seekware.SeekUtility.Temperature.Unit.KELVIN
```

Kelvin (K)

The documentation for this enum was generated from the following file:

- seekware/src/main/java/com/thermal/seekware/SeekUtility.java

Index