

包含功能

- 连接机器人
- 配网功能
- PRP编程
- 透传接口

接入步骤

1.初始化

继承AlphaMiniApplication后,在Manifest声明

```
public class DemoApplication extends AlphaMiniApplication {  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        AlphaMiniSdk.instance().init(this,  
            AlphaMiniParam.newBuilder()  
                .AppId("7nuVS4Hj")  
                .AppKey("1ba2648dd4615916c1c7a4079186d524fa5a4ac5")  
                .build());  
    }  
}
```

2.连接机器人

可以输入机器人SN后四位，连接机器人

```
AlphaMiniSdk.router().toConnBle(this);
```

3.配网

机器人连接后可以对机器人进行配网

```
AlphaMiniSdk.router().toBleConfigNet(this);
```

4.PRP编程

- 进入PRP功能，如果机器人已经连接，可以直接对机器人进行PRP动作编程
- 如果机器人未连接，点击右上角蓝牙按钮连接机器人后，可以对机器人进行PRP动作编程

```
AlphaMiniSdk.router().toPrp(this);
```

5.透传接口

透传接口机器人连接后才可以调用，使用案例参考Demo

```
AlphaMiniSdk.instance().sendData("skillPath", "skillName", "content", new  
    IAlphaMiniCallback<String>() {  
        @Override  
        public void onSuccess(String s) {
```

```

        Log.d("MainActivity", "s: " + s);
    }

    @Override
    public void onError(String s, int i) {
        Log.d("MainActivity", "onError: " + s + ", i: " + i);
    }
});

```

参数说明

- **skillPath** : 技能路径
- **skillName** : 技能名
- **content** : 自定义消息内容

6.添加公共库依赖

```

dependencies {
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'

    configurations.all {
        resolutionStrategy.force 'androidx.appcompat:appcompat:1.2.0'
        resolutionStrategy.force 'androidx.recyclerview:recyclerview:1.0.0'
        resolutionStrategy.force 'androidx.constraintlayout:constraintlayout:2.0.4'
    }

    implementation rootProject.ext.deps.rxbinding
    implementation rootProject.ext.deps.kotlinStdlib
    implementation rootProject.ext.deps.kotlinReflect

    implementation(rootProject.ext.deps.ARouterApi) {
        exclude group: 'com.android.support', module: 'support-media-compat'
    }

    implementation rootProject.ext.deps.protobufLava
    implementation rootProject.ext.deps.multidex
    implementation rootProject.ext.deps.roomRuntime
    implementation rootProject.ext.deps.supportAnnotations
    implementation rootProject.ext.deps.lifecycleExtensions
    implementation rootProject.ext.deps.lifecycleRuntime
    implementation rootProject.ext.deps.mediaCompat
    //implementation rootProject.ext.deps.bugly
    implementation rootProject.ext.deps.palette
    //dagger注入相关
    implementation rootProject.ext.deps.dagger
    implementation rootProject.ext.deps.daggerAndroid
    implementation rootProject.ext.deps.fastjson

    implementation rootProject.ext.deps.eventbus
    implementation rootProject.ext.deps.guava
    implementation rootProject.ext.deps.zxing
    implementation rootProject.ext.deps.glide
    implementation rootProject.ext.deps.webpdecoder

    implementation rootProject.ext.deps.retrofit2
    implementation rootProject.ext.deps.retrofit_gson
    implementation rootProject.ext.deps.rx_retrofit
    implementation rootProject.ext.deps.rxAndroid
    implementation rootProject.ext.deps.okhttp
    //implementation 'com.aliyun.dpa:oss-android-sdk:+'
    implementation 'io.netty:netty-all:4.1.42.Final'

```

```
kapt rootProject.ext.deps.dagger_compiler
kapt rootProject.ext.deps.daggerAndroidProcessor
kapt rootProject.ext.deps.roomAnnotationProcessor
kapt rootProject.ext.deps.ARouterCompiler
kapt rootProject.ext.deps.lifecycleCompiler

compile(name: 'alpha-mini-oversea-edu-sdk-0.0.1-new', ext: 'aar')
}
```

下面的三个androidx maven库要强制固定版本为下面的三个版本

```
configurations.all {
    resolutionStrategy.force 'androidx.appcompat:appcompat:1.2.0'
    resolutionStrategy.force 'androidx.recyclerview:recyclerview:1.0.0'
    resolutionStrategy.force 'androidx.constraintlayout:constraintlayout:2.0.4'
}
```