


# KBSW183443 SDK-Android

Android SDKAndroid Android

- 
- 
- 
- 

SDKAndroid


- Android
- SDK

 SDK

- 
- 
- 
- 
- 
- 
- 

Android Reference Application

- - Android Studio 3.1.3
  - Slamware Android SDK: [Android SDK](#)
  - RoboStudio():[Robostudio installer](#)

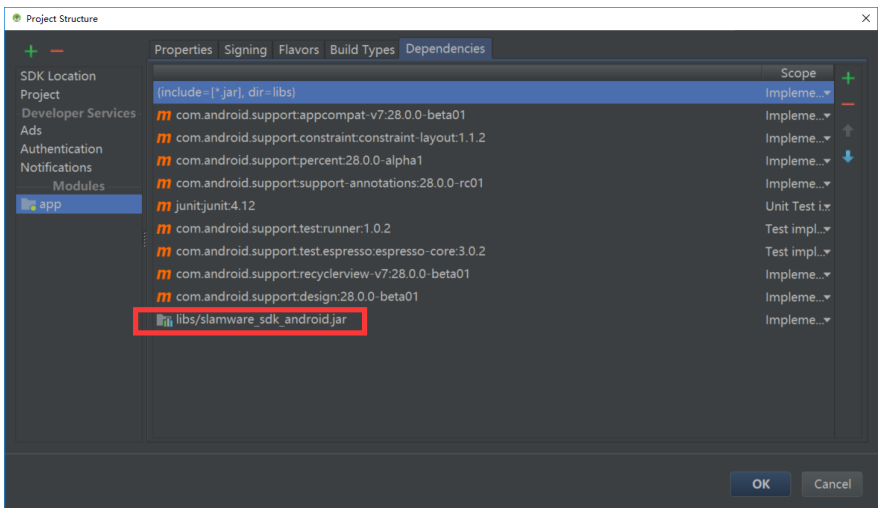
 Android StudioAndroid SDKGradle

- 

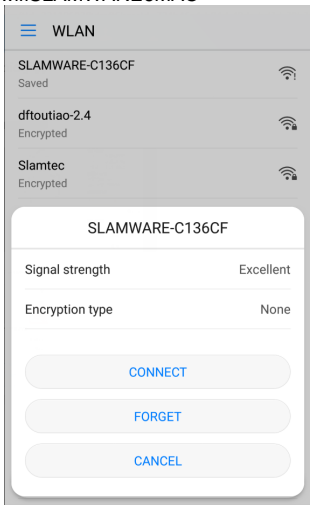
- Slamware SDP mini
- Slamware SDP
- Zeus/Apollo

 Slamware SDKSlamcoreSDK

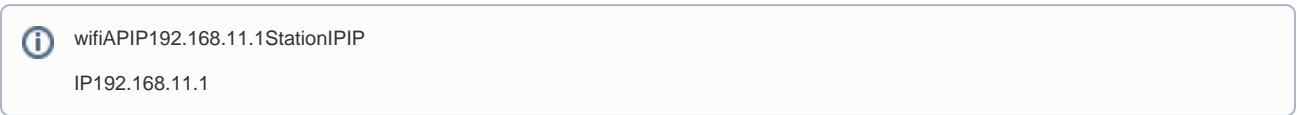
1. Project Structure --> app --> Dependencies SlamwareSDK



2. wifiSLAMWARE6MAC




3. <sup>109</sup>



SDK\_Reference\_Android

机器人IP地址192.168.11.1

连接机器人




1. ARMx86

2. ARM CPUlibrpsdk.soARM v7 ARM v8SDK

2.

4.



当前位置

X: -0.2840757

Y: -0.02219989

YAW: -1.9878426

机器人状态

当前电量 99

运动状态 ERROR

设置目标点

X:  Y:

到这去

运动控制

向前

向左

停止

向右

向后

a.



b. XYXY

c.

1.

a.

moveBy""

```

// go forward
int delayTime = 300;
button_forward.setLongClickRepeatListener(new LongClickButton.LongClickRepeatListener() {
    @Override
    public void repeatAction() {
        try {
            moveAction = robotPlatform.moveBy(MoveDirection.FORWARD);

            System.out.println("repeatAction=====");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}, delayTime);

```

b.

```

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(targetX.length()==0 || targetY.length()==0) {
            Toast.makeText(MainActivity.this, "", Toast.LENGTH_SHORT).show();
        } else {
            try {
                float x = Float.parseFloat(targetX.getText().toString());
                float y = Float.parseFloat(targetY.getText().toString());

                MoveOption moveOption = new MoveOption();
                moveOption.setPrecise(true);
                moveOption.setMilestone(true);

                Log.d(TAG, "Move To");

                moveAction = robotPlatform.moveTo(new Location(x, y, 0), moveOption, 0);
                action.waitForDone();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
});

```

moveTo""

2.

Android RunnableRunnableupdate 100

```

private Handler handler = new Handler();
private Runnable runnable = new Runnable() {
    public void run() {
        this.update();
        handler.postDelayed(this, 100); // 100ms
    }
    ...
}

```

updatePose

```

void update() {
    try {
        /* Pose */
        Pose pose = robotPlatform.getPose();
        current_location_x.setText(Float.toString(pose.getX()));
        current_location_y.setText(Float.toString(pose.getY()));
        current_location_yaw.setText(Float.toString(pose.getYaw()));

        /* */
        int percentage = robotPlatform.getBatteryPercentage();
        current_battery_percentage.setText(Integer.toString(percentage));
        ...
    } catch(Exception e) {
        ...
    }
}

```

BitmapDrawablegetMaSlamcodera dataBitmap ARGB\_8888

```

/* */
int mapWidth = 0;
int mapHeight = 0;

RectF knownArea = robotPlatform.getKnownArea(MapType.BITMAP_8BIT, MapKind.EXPLORE_MAP);
map = robotPlatform.getMap(MapType.BITMAP_8BIT, MapKind.EXPLORE_MAP, knownArea);
mapWidth = map.getDimension().getWidth();
mapHeight = map.getDimension().getHeight();

Bitmap bitmap = Bitmap.createBitmap(mapWidth, mapHeight, ARGB_8888);

for (int posY = 0; posY < mapHeight; ++posY) {
    for (int posX = 0; posX < mapWidth; ++posX) {
        // get map pixel
        byte[] data = map.getData();

        // (-128, 127) to (0, 255)

        int rawColor = data[posX + posY * mapWidth];

        rawColor += 128;

        // fill the bitmap data, by data of B/G/R/A
        bitmap.setPixel(posX, posY, rawColor << 8 | rawColor << 16 | 0xC0 << 24);
    }
}

BitmapDrawable bmpDraw = new BitmapDrawable(bitmap);

imageView.setImageDrawable(bmpDraw);

```



100BitmapDrawable100ms