Reeman 3128 Hardware access documentation (Android)

Document version number:2020/6/16

1.Preparatory Work

1. **Hardware:**
   Reeman Robot,Reeman Development Platform
2. **Software:**
   ReemanSDK,eclipse/Android Studio

2.Initial import

Put the ReemanSDK directly in the libs directory
eclipse：

Android Studio：
1.ReemanSDK.jar Copy to the project libs directory
2.add it to the gradle file

“ implementation files('libs/ReemanSDK.jar') ”

```
dependencies {
    implementation files('libs/ReemanSDK.jar')
```

3.SDK internal controller description

There are two main types of SDK internal controllers, ConnectServer and RobotActionProvider, which is separate from the SDK.
ConnectServer provides peripheral controllers, including navigation, printer, and identity card card reader.Register the corresponding listener callback method with ConnectServer Obtain peripheral real-time data;
RobotActionProvider provides robot hardware control interface and robot system
The system interface.Through the interface provided by RobotActionProvider, walking can be realized.For navigation and serial port operations, be sure to initialize the ConnectServer, and Register the callback function.

4.Peripheral control integration

0.Connectserver initialization

Controller:ConnectServer

Explain:Machine peripheral control object

Method:getInstance(Application s, RscServiceConnectionImpl l)

Method description:Initialize, get the connectserver instance, and control the machine peripherals

Parameter:application、connection call-back

Returned value:ConnectServer, no null value exists

Example:

```
ConnectServer cs = ConnectServer.getInstance(getApplication(), impl);
private RscServiceConnectionImpl impl = new RscServiceConnectionImpl() {
public void onServiceConnected(int name) {
   if (cs == null)
      return;
   if (name == ConnectServer.Connect_D3) {
      cs.register3DSensorListener(Dlistener);
   } else if (name == ConnectServer.Connect_Pr_Id) {
      Log.v("demo", "Connect_Pr_Id ");
      System.out.println("sdk version: " + cs.getSDKVersion());
      cs.registerIDListener(Ilistener);
      }
   }
   public void onServiceDisconnected(int name) {
         System.out.println("onServiceDisconnected.....");
   }
}
```

Rscserviceconnectionimpl is the service binding callback. Here, some peripheral callbacks will be registered. Later, we will Detailed introduction.It is recommended to use in oncreate of activity

Controller:ConnectServer

Explain:Machine peripheral control object

Method:getInstance(Application s）

Method description:Initialize, get the connectserver instance, and control the

machine peripherals

Parameter: application

Returned value:ConnectServer, no null value exists

Example:

ConnectServer cs = ConnectServer.getInstance(getApplication());

Controller:ConnectServer

Explain:Machine peripheral control object

Method:getInstance(）

Method description: Obtain the connectserver instance and control the machine peripherals. Before use, the above two methods are used

One initialized

Parameter: application

Returned value:ConnectServer,There is a null value

Example: ConnectServer cs = ConnectServer.getInstance();

5. Hardware and system interface integration

0. RobotActionProvider initialize

Controller：RobotActionProvider

Explain：Machine hardware and system interface control object

Method：getInstance()

Method description： Initialize, get the RobotActionProvider instance, and control the machine peripherals

Parameter：None

Returned value:RobotActionProvider, no null value

Example:

RobotActionProvider.getInstance().systemHome();

1. Serial port 3 communication

Method name：sendTtyS3(byte[] d)

Method description： The internal method of RobotActionProvider provides serial port 3 operation access (see the appendix serial port communication Agreement)

Parameter：byte array

Returned value:None

2. Serial port 4 communication

Method name：sendTtyS4(byte[] data)

Method description： The internal method of RobotActionProvider provides serial port 4 operation access (see the appendix serial port communication Communication protocol)

Parameter：byte array

Byte input rule:[len,cmd,p1,p2…,pn,time]

Returned value:None

Head control and arm control can be called by referring to serial port protocol

Example:

Eye light control byte array:

byte[] in_ear = { 0X03, (byte) 0X80, (byte) codeP, 0 };

3. Get the high 8 bits of short

Method name：getHight8(short h)

Method description： The internal method of RobotActionProvider provides short high-8-bit acquisition. In the communication protocol of serial port 4, some of the high-8-bit input is needed to cooperate with the serial port protocol

Parameter：Short type

Returned value:byte

Example:

byte h = RobotActionProvider.getInstance().getHight8((short) -130);

4. Get the lower 8 bits of short

Method name：getLow8(short l)

Method description： The internal method of RobotActionProvider provides short type low 8-bit acquisition. In the communication protocol of serial port 4, some of the low 8-bit needs to be input for use with the serial port protocol

Parameter： Short type

Returned value:byte

Example:

byte l = RobotActionProvider.getInstance().getLow8((short) -130);

5. Execute adb instruction

Method name： exec(String param)

Method description： RobotActionProvider internal method, providing system operation permission

Parameter： ADB shell instruction

Returned value:None

6. Forward (with speed control)

Interface name： moveFront(int param, int speed)

Interface description： Provided internally by the RobotActionProvider,Control the machine forward

Parameter：

param:Forward distance, value range greater than 1, unit: cm

speed： The default value is 200, and the value range is (0 < speed < = 600)

Note: speed control is not supported on navigation version machines; forward side is supported on non navigation version machines Direction speed control. Illegal speed machine will not move forward

Returned value:None

7. Forward (default speed)

Interface name： moveFront(int param)

Interface description： Provided internally by RobotActionProvider,Control the machine forward

Parameter：

param:Forward distance, value range greater than 1, unit: cm

Speed defaults to 200

Returned value:None

8. Back off

Interface name：moveBack(int param, int speed)

Interface description： Provided by the RobotActionProvider control the machine backward

Parameter：

param:Backward distance, value range greater than 1, unit: cm

speed:The default value is 200, and the speed is not changed; control is not supported temporarily

Returned value:None

9. Turn left

Interface name：moveLeft(int param, int speed)

Interface description： Provided internally by the RobotActionProvider,Control the machine to turn left

Parameter：

param:Value range 1-360 degrees

speed:the default value is 200, do not change the speed; control is not supported temporarily

Returned value:None

10. Turn right

Interface name：moveRight(int param, int speed)

Interface description： Provided internally by the RobotActionProvider,Control the machine to turn right

Parameter：

param:Value range 1-360 degrees

speed:the default value is 200, do not change the speed; control is not supported temporarily

Returned value:None

11. Stop the front and rear left and right movements of the base

Interface name：stopMove()

Interface description：Provided internally by the RobotActionProvider,

Parameter：None

Returned value:None

12. Get device unique ID

Interface name：getRobotID()

Interface description：Provided internally by the RobotActionProvider,

Parameter：None

Returned value:String

13. Home key function of shielding system

Interface name：shieldHome(int d)

Interface description：Provided internally by the RobotActionProvider,Home key of direct simulation system

Parameter：0 shielded; 1 unshielded

Returned value:None

14. Get SDK version number

Interface name：getSDKVersion()

Interface description： Provided internally by the RobotActionProvider,Get SDK Version (date)

Parameter：None

Returned value:String

15. Get MCU version number

Interface name：getHardVersion(int t)

Interface description： Provided internally by the RobotActionProvider,Get MCU version number

Parameter：1. Base; 2. Chest plate

Returned value:String version number

16. Detection and reception of emergency stop switch

Interface name：getScramState()

Interface description：Provided internally by the RobotActionProvider,Active access to emergency stop switch status

The system will also report the status of emergency stop switch through broadcast, see "broadcast description"

Parameter：None

Returned value: int 0 Press lower,1 open

17. Get base version

Interface name：getBottomMode()

Interface description：Provided internally by the RobotActionProvider,Whether the active acquisition base supports navigation, laser navigation and visual navigation classification

Parameter：None

Returned value:0 ordinary,1 Laser navigation,2 Visual navigation

18. Return the value of machine ultrasonic infrared

Interface name：getRevTtys3(int a)

Interface description： Provided internally by the RobotActionProvider,Take the initiative to obtain the value of ultrasonic infrared returned to the machine, and provide the data acquisition of ultrasonic infrared module. The user needs to take the initiative to obtain the module value, and the interval time of each time is not less than 100ms

Parameter：int type

21: right infrared remote control code data 22: left infrared remote control code data 23: FCC infrared remote control code data

24: Top infrared remote control code data 25: rear ultrasonic distance 26: front

ultrasonic distance

27: left ultrasonic distance 28: left middle ultrasonic distance 29: middle ultrasonic distance

30: right middle ultrasonic distance 31: right ultrasonic distance

Returned value:int type Feedback on behalf of each module

6. Navigation module

0. Execute navigation related instructions

Method name:sendRosCom(String c)
Method explain:RobotActionProvider internal method,As navigation control entry
Parameter:
c:Navigation control command
Returned value:None
Example of navigation byte array:
sendRosCom ("goal:nav[A]")

1. Receive navigation related feedback

Method name:registerROSListener(OnROSListener listener)
Method explain:Connectserver internal methods,Provide navigation module data reception
Listener, used to receive data returned by ROS
Parameter:
listener:Callback listener
Returned value:None
Tips:Specific callback,please refer to Reeman ROS Serial protocol description.pdf