

cVendServiceSample

cVendServiceSample 提供了 Ethernet Tethering、BCR GPIO Pin、OneWire External 等功能的测试 demo，并提供了对应 api 的具体使用方法。

notes

要使用 cVendServiceSample 需要先安装 cVendService，需要手动执行如下指令启动 service，若需要开机自启动，需要将 service 预置到 bsp 中

```
adb shell am startservice -n
com.advantech.cvendservice/com.advantech.cvendservice.CVendService
```

cVendService

cVendService 是一个用于控制 Ethernet Tethering、BCR GPIO Pin、OneWire External 于一体的 service，对外暴露了广播接口，其收到对应的 action 就会去处理相应的动作，并且返回结果。

EthernetTethering

- 接口说明

```
// action on ethernet tethering
private static final String ACTION_ETHETHERING_ON =
"com.advantech.aim75.ETHETHERING_ON";

// action off ethernet tethering
private static final String ACTION_ETHETHERING_OFF =
"com.advantech.aim75.ETHETHERING_OFF";

// action return ethernet tethering status
private static final String ACTION_ETHETHERING_STATUS =
"com.advantech.aim75.ETHETHERING_STATUS";

// action return ethernet tethering status key
private static final String IS_ETHETHERING_ON = "is_ethethering_on";
```

- 使用说明

```
// 使用 sendBroadcast 发送广播，进而打开、关闭 Ethernet Tethering

// on
sendBroadcast("com.advantech.aim75.ETHETHERING_ON");

// off
sendBroadcast("com.advantech.aim75.ETHETHERING_OFF");
```

- 返回值说明

```
// 要获取返回值，需要注册广播

IntentFilter filter = new IntentFilter();
```

```

filter.addAction("com.advantech.aim75.ETHTETHERING_STATUS");
registerReceiver(broadcastReceiver, filter);

// 创建广播接收器
private final BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if ("com.advantech.aim75.ETHTETHERING_STATUS".equals(action)) {
            // return status
            boolean status = intent.getBooleanExtra("is_ethtethering_on", false);
            Log.d(TAG, "status : " + status);
        }
    }
};

```

BCRGPIOPin

- 接口说明

```

// deactivate
private static final String ACTION_BCRPIN_HIGH =
"com.advantech.aim75.BCRPIN_HIGH";

// active
private static final String ACTION_BCRPIN_LOW = "com.advantech.aim75.BCRPIN_LOW";

```

- 使用说明

```

// // sendBroadcast deactivate action
sendBroadcast("com.advantech.aim75.BCRPIN_HIGH");

// sendBroadcast active action
sendBroadcast("com.advantech.aim75.BCRPIN_LOW");

```

- 返回值说明

None

OnewireExternal

- 接口说明

```
// 获取 DockingName 和 DockingValue 的广播接口
private static final String ACTION_ONEWIRE = "com.advantech.aim75.ONEWIRE";

// 获取返回值的广播接口
private static final String ACTION_ONEWIRE_RESULT =
"com.advantech.aim75.ONEWIRE_RESULT";

// 携带参数指令的 key
private static final String ONEWIRE_RESULT_KEY = "apccmd";

// 携带返回值的 key
private static final String ONEWIRE_RESULT = "ReturnData";
```

- 使用说明

```
// 发送 ACTION_ONEWIRE 并携带 apccmd 80,
Intent intent = new Intent();
intent.setAction("com.advantech.aim75.ONEWIRE");
intent.putExtra("apccmd", "80");
sendBroadcast(intent);

// 发送 ACTION_ONEWIRE 并携带 apccmd 81,
Intent intent = new Intent();
intent.setAction("com.advantech.aim75.ONEWIRE");
intent.putExtra("apccmd", "81");
sendBroadcast(intent);
```

- 返回值说明

```
IntentFilter filter = new IntentFilter();
filter.addAction("com.advantech.aim75.ONEWIRE_RESULT");
registerReceiver(broadcastReceiver, filter);

private final BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (ACTION_ONEWIRE_RESULT.equals(action)) {
            // result
            String result = intent.getStringExtra("ReturnData");
        }
    }
};
```