## 无Tx或无Rx的检测方案适配问题

以无Tx为例，云端下发检测方案脚本如下：

{

"testType": null,

"config": [

{

"checkId": "00bf7710-6f75-11f0-b605-fd41a42e4a85",

"detectionItem": "胎压系统监控检测",

"executions": 5,

"cycleEndStepIdx": 1,

"delayTime": 1000,

"isNotifyCollector": true,

"continueIgnoreFail": 1,

**"txCans": [],**

"rxCans": [

{

"checkId": "6a8e68e0-6f75-11f0-b605-fd41a42e4a85",

"checkTitle": "Step2:胎压故障监测",

"signalName": [

"ZCUL\_TPMS\_NotCalibration",

"ZCUL\_TPMS\_FLWarnPresLow",

"ZCUL\_TPMS\_FLWarnPresHigh"

],

"signalId": [

"4501130715185",

"4501130715188",

"4501130715189"

],

"value": [

"0",

"0",

"0"

],

"pollingFrequency": 0,

"canId": [

"0x418",

"0x418",

"0x418"

],

"delayTime": 0

}

],

"initCans": [],

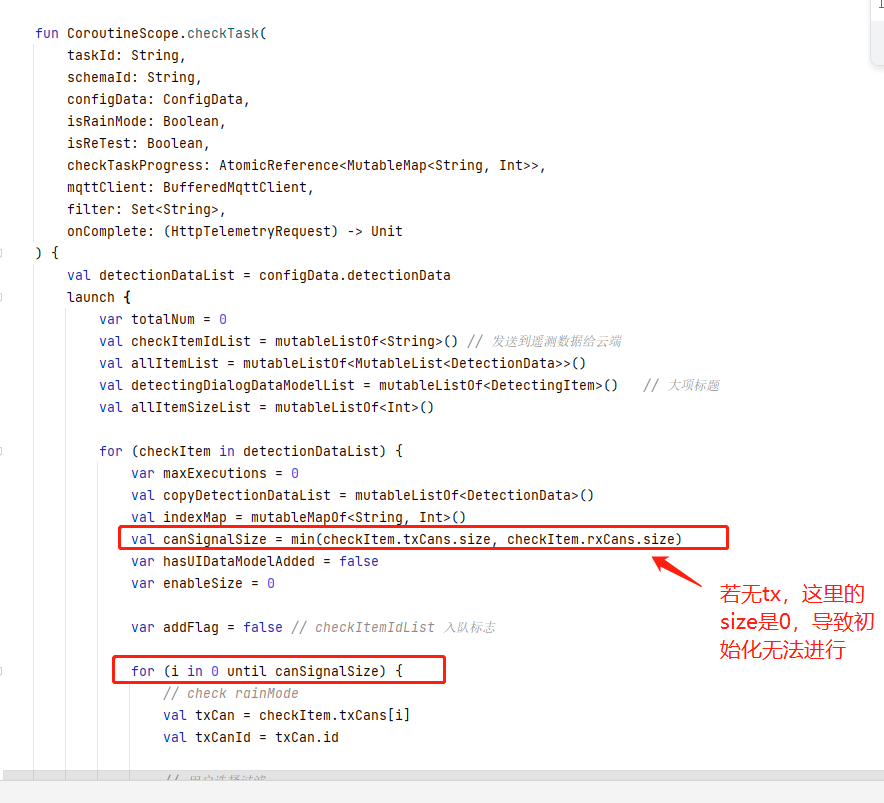
"isParallel": false

}

]

}

其中**txCans**是空数组，这种情况下安卓无法执行检测方案，原因是安卓代码里：



安卓本意是让tx和rx的数量对齐，现在需要满足tx为空的情况，需修改安卓代码适配。

## DTC步骤化需求：

需三端一起开发，链路： 云端 -> 安卓 -> 下位机 -> 云端

### 云端发送dtc配置信息（以检测项为单位）

修改原config接口结构，

原config数组的item新增List<DetectionCmd> beforeCmds, List<DetectionCmd> afterCmds,

两个元素，其中DetectionCmd结构如下：

{

“method”: “CleanDTC”,

“checkId”: “uuid”,

“state”: 0

}, 其中mehodName表示执行的cmd名称，checkId是检测项id, state 0表示开始，1表示结束。

整个config如下：

{

"testType": null,

"config": [

{

"checkId": "00bf7710-6f75-11f0-b605-fd41a42e4a85",

"detectionItem": "胎压系统监控检测",

"executions": 5,

"cycleEndStepIdx": 1,

"delayTime": 1000,

"isNotifyCollector": true,

"continueIgnoreFail": 1,

**“beforeCmds”: [**

**{**

**“method”: “CleanDTC”,**

**“checkId”: “**00bf7710-6f75-11f0-b605-fd41a42e4a86**”,**

**“state”: 0**

**}**

**].**

**“afterCmds”: [],**

"txCans": [],

"rxCans": [

{

"checkId": "6a8e68e0-6f75-11f0-b605-fd41a42e4a85",

"checkTitle": "Step2:胎压故障监测",

"signalName": [

"ZCUL\_TPMS\_NotCalibration",

"ZCUL\_TPMS\_FLWarnPresLow",

"ZCUL\_TPMS\_FLWarnPresHigh"

],

"signalId": [

"4501130715185",

"4501130715188",

"4501130715189"

],

"value": [

"0",

"0",

"0"

],

"pollingFrequency": 0,

"canId": [

"0x418",

"0x418",

"0x418"

],

"delayTime": 0

}

],

"initCans": [],

"isParallel": false

}

]

}

### 安卓获取配置并下发给下位机：

遇到beforCmds则在该检测项开始前先执行，afterCmds则是检测项结束后执行，下发给下位机的json如下：

{

"method": "CleanDTC",

"test\_id": "xxx",

"state": 0,

“checkId”: uuid

}

### 下位机获取安卓下发的指令执行并返回执行结果给云端：

下位机收到安卓上发的json后执行相关指令，记录指令的开始结束时间以及执行结果。执行结果通过v1/devices/me/eol的 topic 发送，新增key为**dtc\_app**的json，格式如下：

{

**…**

“dtc\_app”: [

{

“methodName”: “cleanDTC”,

“checkId”: uuid,

“state”: 1,

“start\_time”: 17432445254325000,

“end\_time”: 17432445254325000

}

]

}