

Application Note

ADAS 车载摄像头热插拔机制实现



Jiawei Hu

摘要

高级辅助驾驶系统 (ADAS) 通常采用多个摄像头感知外围环境，如果一个摄像头发生故障或者接入，系统需要能够继续依赖其他摄像头工作或者实时接入摄像头数据。热插拔机制允许系统在不关机，不重启的情况下，实时检测到摄像头移除或接入，当检测到摄像头移除时，将其标记为不可用状态，并且保证其他正常摄像头正常使用；当检测到摄像头接入时，系统能实时接入摄像头数据。因此热插拔机制是构建高安全性和高可用性 ADAS 不可或缺的一环。

ADAS FPD-Link 作为车载摄像头重要器件，其丰富诊断功可帮助用户实现监控摄像头接入或者移除，从而实现热插拔机制。本文分别讲述 FPD-Link 器件热插拔机制相关诊断功能以及相关参考代码，进而帮助用户快速实现热插拔机制。

内容

1 诊断机制.....	2
1.1 Link lock 检测.....	2
1.2 Video pass 检测.....	2
1.3 使能或关闭 RX forwarding.....	2
1.4 清除解串器错误状态寄存器.....	2
1.5 关闭 DFE 模块.....	2
1.6 Reset SER/Sensor 配置以及使能 DFE 模块.....	2
2 系统热插拔机制流程图.....	4
3 参考代码.....	5
4 参考文献.....	6

1 诊断机制

1.1 Link lock 检测

对于串化解串器是否成功建立 link，解串器可通过寄存器 0x4D bit0 判断解串器是否与远端串化器建立 link。

```
board.Writel2C(DesAddr,0x4c,0x01/12/24/38) # Select RX0/1/2/3
```

check 0x4D bit 0 是否为 1 判断是否与 RX0/1/2/3 串化器 lock

1.2 Video pass 检测

解串器可通过寄存器 0x7D 设置 video pass 判断标准以及通过寄存器 0x4D bit1 判断是否收到有效 video，本文已 0x7D 设置为 0x33 为例（解串器收到稳定分辨率图像以及收到 3 帧数据之后，video pass 置 1）。

```
board.Writel2C(DesAddr,0x4c,0x01/12/24/38) # Select RX0/1/2/3
```

```
board.Writel2C(DesAddr,0x7D,0x33) # disable watchdog and set pass threshold
```

check 0x4D bit 1 是否为 1 判断是否 video pass

1.3 使能或关闭 RX forwarding

当通过 Link lock 检测到摄像头接入或者移除时，解串器可通过寄存器 0x20 bit4-7 分别使能或者关闭 RX0/1/2/3 数据转发。

1.4 清除解串器错误状态寄存器

当发生摄像头热插拔时，需要对相关诊断寄存器进行读清错误操作。

```
board.Writel2C(DesAddr,0x4c,0x01/12/24/38) # Select RX0/1/2/3
```

```
board.Readl2C(DesAddr,0x4D) # Clear status register of errors
```

```
board.Readl2C(DesAddr,0x4E) # Clear status register of errors
```

```
board.Readl2C(DesAddr,0x55) # Clear status register of errors
```

```
board.Readl2C(DesAddr,0x56) # Clear status register of errors
```

```
board.Readl2C(DesAddr,0x7A) # Clear status register of errors
```

```
board.Readl2C(DesAddr,0x7B) # Clear status register of errors
```

1.5 关闭 DFE 模块

检测到摄像头移除时，需要关闭解串器 DFE 模块。

```
board.Writel2C(DesAddr, 0xB1,0x71)
```

```
board.Writel2C(DesAddr, 0xB2,0x20)
```

```
board.Writel2C(DesAddr, 0xB1,0x90)
```

```
board.Writel2C(DesAddr, 0xB2,0x00)
```

1.6 Reset SER/Sensor 配置以及使能 DFE 模块

#Hard reset SER

```
reg_0x58 = board.Readl2C(DesAddr,0x58)
```

```
reg_0x58 = reg_0x58 | 0x60 # Enable I2C Passthrough with auto ACK
```

```
board.Writel2C(DesAddr,0x58,reg_0x58)
```

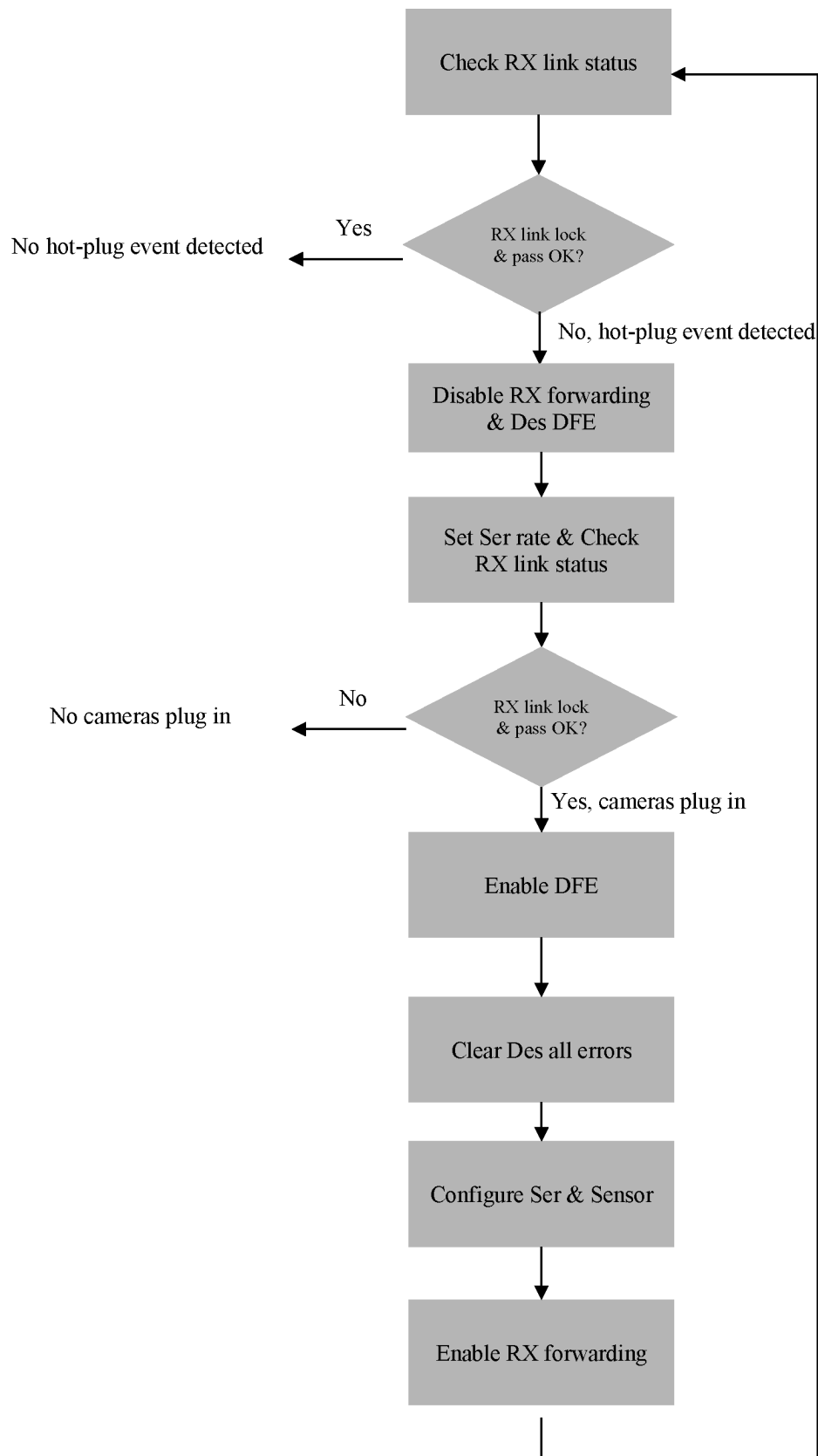
```
board.Writel2C(serAlias[rx_port],0x01,0x02) # Ser Hard Reset
```

```
time.sleep(0.01)#wait 10ms
```

```
board.Writel2C(serAlias[rx_port],0x0A,0x12)
```

```
reg_0x58 = reg_0x58 & 0x5F # Disable auto ACK and I2C passthrough  
board.Writel2C(DesAddr,0x58,reg_0x58)  
board.Writel2C(DesAddr, 0xB1,0x90)  
board.Writel2C(DesAddr, 0xB2,0x40)  
board.Writel2C(DesAddr, 0xB1,0x71)  
board.Writel2C(DesAddr, 0xB2,0x20)  
time.sleep(0.0001)#Wait 100us  
board.Writel2C(DesAddr, 0xB2,0x00)  
#Re-configure sensor
```

2 系统热插拔机制流程图



3 参考代码

```
def CheckRXLinkStatus(rx_port):
    select_rx_port(rx_port)
    port_status1 = board.ReadI2C(DesAddr,0x4D)
    port_status2 = board.ReadI2C(DesAddr,0x4E)
    parity_err0 = board.ReadI2C(DesAddr,0x55)
    parity_err1 = board.ReadI2C(DesAddr,0x56)
    csi_err = board.ReadI2C(DesAddr,0x7A)
    csi_err_cnt = board.ReadI2C(DesAddr,0x7B)
    port_status1_pass = port_status1 == (rx_port << 6) + 3
    port_status2_pass = port_status2 == 4
    parity_err_pass = ((parity_err0*256) + parity_err1) < 1
    csi_err_pass = csi_err == 0 and csi_err_cnt == 0
def clearDesAllErrors(rx_port):
    board.ReadI2C(DesAddr,0x4D) # Clear status register of errors.
    board.ReadI2C(DesAddr,0x4E) # Clear status register of errors.
    board.ReadI2C(DesAddr,0x55) # Clear status register of errors.
    board.ReadI2C(DesAddr,0x56) # Clear status register of errors.
    board.ReadI2C(DesAddr,0x7A) # Clear status register of errors.
    board.ReadI2C(DesAddr,0x7B) # Clear status register of errors.
print "Port", str(rx_port), "Clear status register of errors 0x4D, 0x4E, 0x55, 0x56, 0x7A, and 0x7B"
def recovery_SER(rx_port):
    select_rx_port(rx_port)
    board.WriteI2C(DesAddr, 0xB1,0x71)
    board.WriteI2C(DesAddr, 0xB2,0x20)
    board.WriteI2C(DesAddr, 0xB1,0x90)
    board.WriteI2C(DesAddr, 0xB2,0x00)

    #Hard reset SER
    reg_0x58 = board.ReadI2C(DesAddr,0x58)
    reg_0x58 = reg_0x58 | 0x60 # Enable I2C Passthrough with auto ACK
    board.WriteI2C(DesAddr,0x58,reg_0x58)
    board.WriteI2C(serAlias[rx_port],0x01,0x02) # Ser Hard Reset
    time.sleep(0.01)#wait 10ms
    if SerConfig[rx_port] == "971":
        board.WriteI2C(serAlias[rx_port],0x4B,0x02) # disable BC alternate mode auto detect
        board.WriteI2C(serAlias[rx_port],0x49,0x06) # decrease link detect timer on 971 BC-RX
        print 'writing 0x4B and 0x49 on 971'
    board.WriteI2C(serAlias[rx_port],0x0A,0x12)
    reg_0x58 = reg_0x58 & 0x5F # Disable auto ACK and I2C passthrough
    board.WriteI2C(DesAddr,0x58,reg_0x58)

    board.WriteI2C(DesAddr, 0xB1,0x90)
    board.WriteI2C(DesAddr, 0xB2,0x40)
    board.WriteI2C(DesAddr, 0xB1,0x71)
    board.WriteI2C(DesAddr, 0xB2,0x20)
    time.sleep(0.0001)#wait 100us
    board.WriteI2C(DesAddr, 0xB2,0x00)

    print '\nPort', str(rx_port), 'recovery loop complete, checking status:'
    clearDesAllErrors(rx_port)
    time.sleep(0.1) #wait time for error accumulation
CheckRXLinkStatus(rx_port)
Cycle = 1000
print "Hot plug testing - remove camera when ready"
for i in range(0,cycle):
    RX_PORT_STS1 = board.ReadI2C(DesAddr, 0x4D)
    if (RX_PORT_STS1 != 0x3):
        print "Hot plug event detected"
        break
    elif i == cycle-1:
        print "No hot plug event detected"
for i in range(0,cycle):
    RX_PORT_STS1 = board.ReadI2C(DesAddr, 0x4D)
    if (RX_PORT_STS1 == 0x3):
        print "completing hot plug"
        recovery_SER(0) #note that need reconfigure SER & Sensor
        break
    elif i == cycle-1:
        print "lock not re-gained within expected timeframe"
```

4 参考文献

- [DS90UB960-Q1 data sheet \(SNLS589C\)](#)

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
版权所有 © 2025，德州仪器 (TI) 公司