勘误附图

1、清单 36.3（第 260 页）少了 4 个左花括号，比较值错误，如下图所示：

```
//端点2输出回调函数（从主机到设备）
void EP2_OUT_Callback(void)
{
  uint8_t Receive_Buffer;                          //保存输出报告中的字段（1个字节）
  USB_SIL_Read(EP2_OUT, &Receive_Buffer);          //从接收包缓冲区中读出一个字节

  if (Receive_Buffer&0x1) {                //判断D1需要设置的状态（0为熄灭，1为点亮）
    STM_EVAL_LEDOff(LED_D1);                       //设置引脚为低电平（点亮LED_D1）
  } else {
    STM_EVAL_LEDOn(LED_D1);                        //设置引脚为高电平（熄灭LED_D1）
  }

  if (Receive_Buffer&0x1) {                //判断D2需要设置的状态（0为熄灭，1为点亮）
    STM_EVAL_LEDOff(LED_D2);                       //设置引脚为低电平（点亮LED_D2）
  } else {
    STM_EVAL_LEDOn(LED_D2);                        //设置引脚为高电平（熄灭LED_D2）
  }

  if (Receive_Buffer&0x1) {                //判断D3需要设置的状态（0为熄灭，1为点亮）
    STM_EVAL_LEDOff(LED_D3);                       //设置引脚为低电平（点亮LED_D3）
  } else {
    STM_EVAL_LEDOn(LED_D3);                        //设置引脚为高电平（熄灭LED_D3）
  }

  if (Receive_Buffer&0x1) {                //判断D4需要设置的状态（0为熄灭，1为点亮）
    STM_EVAL_LEDOff(LED_D4);                       //设置引脚为低电平（点亮LED_D4）
  } else {
    STM_EVAL_LEDOn(LED_D4);                        //设置引脚为高电平（熄灭LED_D4）
  }

  SetEPRxStatus(ENDP2, EP_RX_VALID);           //处理完后再次设置接收状态为VALID
}
```

0x2

0x4

0x8

2、清单 43.2（第 313 页倒数第 6 行），删除判断语句中的“**&& RequestNo == GET_PROTOCOL**”，更新后如下图所示

```
  } else if ((Type_Recipient == (CLASS_REQUEST | INTERFACE_RECIPIENT))) {
    switch(RequestNo) {
      case GET_PROTOCOL: {
        CopyRoutine = Joystick_GetProtocolValue;                    //获取协议值
        break;
      }
      case SET_REPORT: {
        CopyRoutine = Joystick_SetReport;                           //设置报告
        break;
      }
      case GET_REPORT:{
        CopyRoutine = Joystick_GetReport;                           //获取报告
        break;
      }
      default:break;
    }
  }

  if (CopyRoutine == NULL) {
    return USB_UNSUPPORT;
  }
  pInformation->Ctrl_Info.CopyData = CopyRoutine;
  pInformation->Ctrl_Info.Usb_wOffset = 0;
  (*CopyRoutine)(0);
  return USB_SUCCESS;
}
```

3、清单 43.2（第 314 页），Joystick_GetReport 函数最后少截了 3 行代码，Joystick_Status_In

函数中的比较值错误，更新后如下图所示：

```
    return Report_In_Buf;                                    //返回输入报告缓冲区首地址
  }
}

void Joystick_Status_In(void)
{
  if (Report_Out_Buf&0x1) {                  //判断D1需要设置的状态（0为熄灭，1为点亮）
    STM_EVAL_LEDOff(LED_D1);                                 //设置引脚为低电平（点亮LED_D1）
  } else {
    STM_EVAL_LEDOn(LED_D1);                                  //设置引脚为高电平（熄灭LED_D1）
  }

  if (Report_Out_Buf&0x2) {                  //判断D2需要设置的状态（0为熄灭，1为点亮）
    STM_EVAL_LEDOff(LED_D2);                                 //设置引脚为低电平（点亮LED_D2）
  } else {
    STM_EVAL_LEDOn(LED_D2);                                  //设置引脚为高电平（熄灭LED_D2）
  }

  if (Report_Out_Buf&0x4) {                  //判断D3需要设置的状态（0为熄灭，1为点亮）
    STM_EVAL_LEDOff(LED_D3);                                 //设置引脚为低电平（点亮LED_D3）
  } else {
    STM_EVAL_LEDOn(LED_D3);                                  //设置引脚为高电平（熄灭LED_D3）
  }

  if (Report_Out_Buf&0x8) {                  //判断D4需要设置的状态（0为熄灭，1为点亮）
    STM_EVAL_LEDOff(LED_D4);                                 //设置引脚为低电平（点亮LED_D4）
  } else {
    STM_EVAL_LEDOn(LED_D4);                                  //设置引脚为高电平（熄灭LED_D4）
  }
}
```