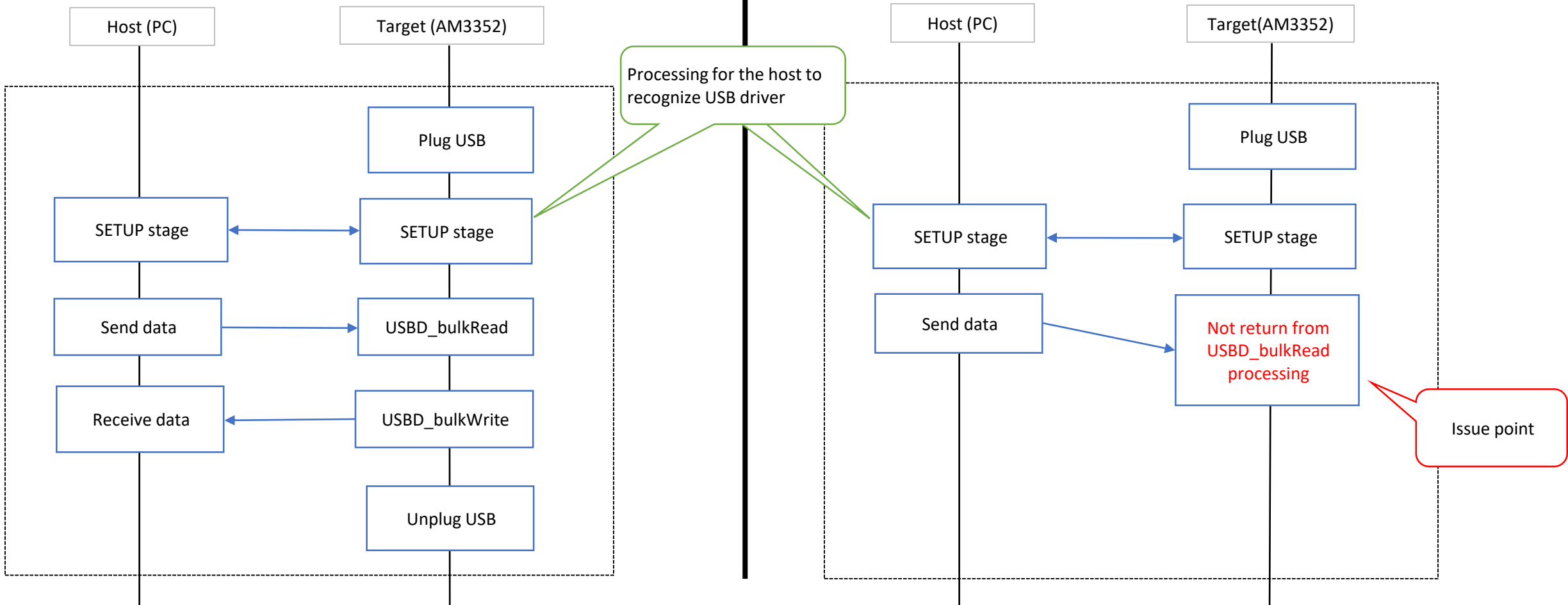


# plug/unplug test overview (Repeat the following test)

Case :Data transfer success

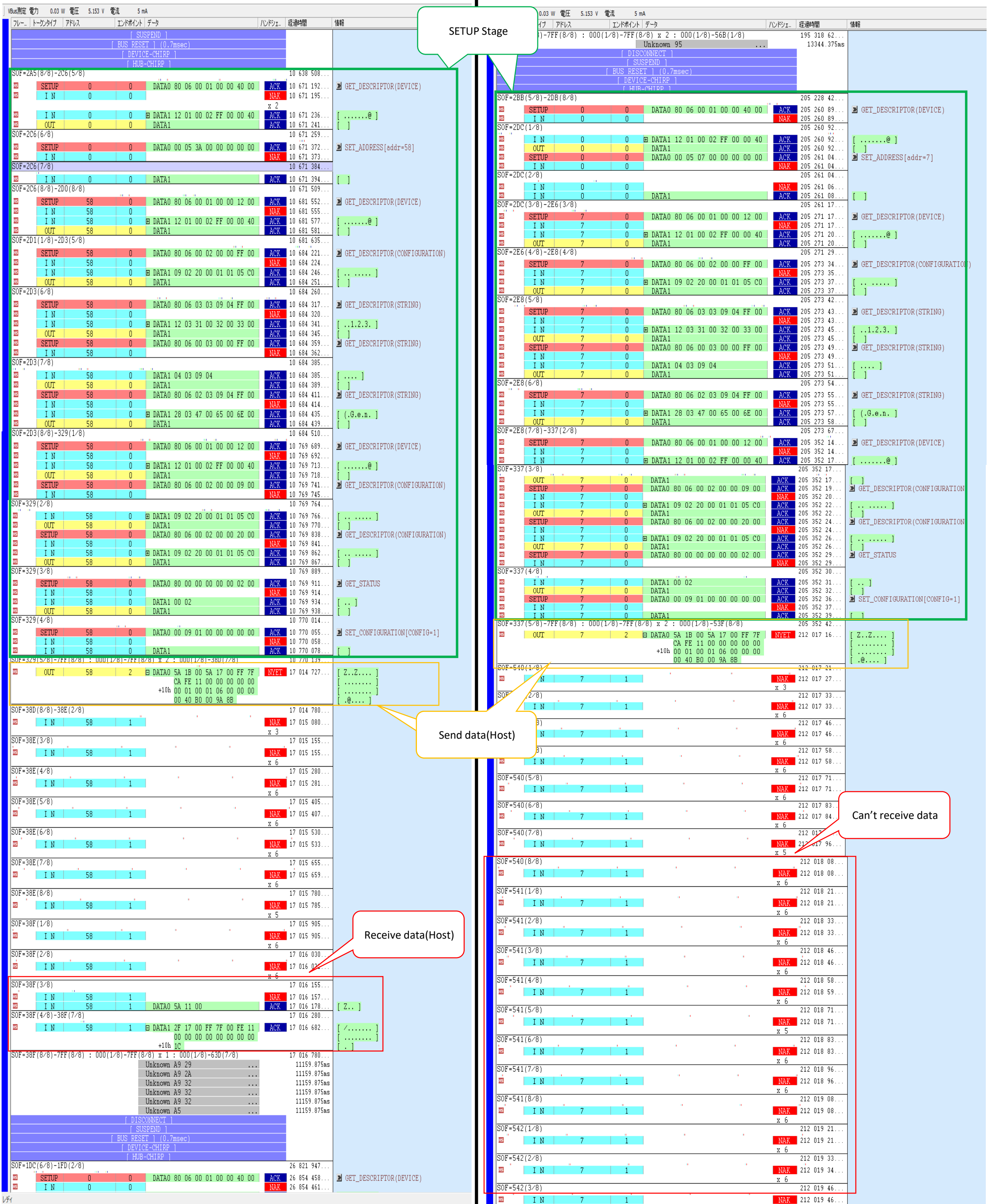
Case :Data transfer fail



## Protocol Overview

success

fail



When suspended, we can confirm that exit () has been reached.

### psPrivateBulkData information

C:\ti\pdk\_am335x\_1\_0\_16\packages\ti\drv\usb\src\usb\_func\device\usbdbulk.c

```

957
958
959 int32_t USBD_bulkRead(USB_Handle handle, uint8_t* buffer, uint32_t* dataSize)
960 {
961     USB_Params* usbParams;
962     tUSBDBulkDevice* pusbBulkDevice;
963     struct usbGadgetObj* pGadgetObj;
964     tBulkInstance* psInst;
965     int32_t size = 0;
966
967     usbParams = handle->usbParams;
968     pusbBulkDevice = (tUSBDBulkDevice*)usbParams->usbClassData;
969     psInst = pusbBulkDevice->psPrivateBulkData;
970
971     /* Setup request for the lower layer OUT transfer */
972     pGadgetObj = (struct usbGadgetObj*)psInst->psDevInfo->pGadgetObj;
973
974     if (psInst->eBulkRxState == BULK_STATE_IDLE)
975     {
976         /* prepare for incoming bulk connection from host */
977         *dataSize = psInst->usLastRxSize = 0;
978
979         /* set state to waiting */
980         psInst->eBulkRxState = BULK_STATE_WAIT_DATA;
981
982         /* Setup request for the lower layer to be ready for OUT transfer */
983         usbSetupReq(pGadgetObj,
984
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

60 B:Var.View %open svBulkInstance
svBulkInstance = (
  @psDevInfo = 0x8003D81C,
  @psGadgetObj = 0x8003D8E4,
  eBulkRxState = BULK_STATE_WAIT_DATA,
  eBulkTxState = BULK_STATE_IDLE,
  usDeferredOpFlags = 0,
  usLastTxSize = 17,
  usLastRxSize = 0,
  bConnected = 1,
  ucInEndpoint = 1,
  ucOutEndpoint = 2,
  ucInterface = 0,
  readSem = 0x0,
  writeSem = 0x0)

```

**eBulkRxState = BULK\_STATE\_WAIT\_DATA**  
Therefore, it is considered that the USBD\_bulkRead function is being executed.

```

200 static tBoolean+
201 ProcessDataFromHost(struct usbGadgetObj *pGadgetObj, +
202 const tUSBDBulkDevice *psDevice, +
203 uint32_t length)
204 {
205     tBulkInstance *psInst;
206
207     /* Get a pointer to our instance data. */
208     psInst = psDevice->psPrivateBulkData;
209
210     /* handle DMA transaction.
211     We got here because we have received a bulk-out interrupt */
212
213     psInst->eBulkRxState = BULK_STATE_IDLE;
214     psInst->usLastRxSize = length;
215
216     usb_osalPostLock(psInst->readSem);
217
218     /* data should be already received. */
219     psDevice->pfnRxCalBack(pGadgetObj, +
220 psDevice->psPrivateBulkData, +
221 psDevice->usLastRxSize);

```

On the other hand, since RxState = BULK\_STATE\_IDLE is not established, you can see that the ProcessDataFromHost function is not executed.

We think there is a problem during the receive process.