Hi

We have an STM32L4x5 project. We are using STM32CubeMX to generate our HAL based framework and have a bunch of hardware all working great. We were noticing application level CRC errors on receipt of data being receiving via a CDC USB connection and this has caused me to create a simple test case which shows it. We are probably doing something wrong here, but would love some insight into this.

```
char output[90] =
"0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567\r\n";
 int index = 0;
 uint8_t * out = (uint8_t*)&output;
 while (1)
 {
  USBD_CDC_HandleTypeDef *hcdc = (USBD_CDC_HandleTypeDef*)hUsbDeviceFS.pClassData;
  while (hcdc->TxState != 0){
   // wait for usb tx done
  }
  if (CDC_Transmit_FS(out+index, 10)== USBD_OK)
  {
   index += 10;
  }
  if (index >= sizeof(output))
  {
   index = 0;
   DelayMilliseconds(50);
  }
  WatchdogReset();
 }
```

This is a complete 10 second hack of a test case which we put in main.c, but generates the same issue, which is on the PC we receive data like this:

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
012345678901234567890**1230**1234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

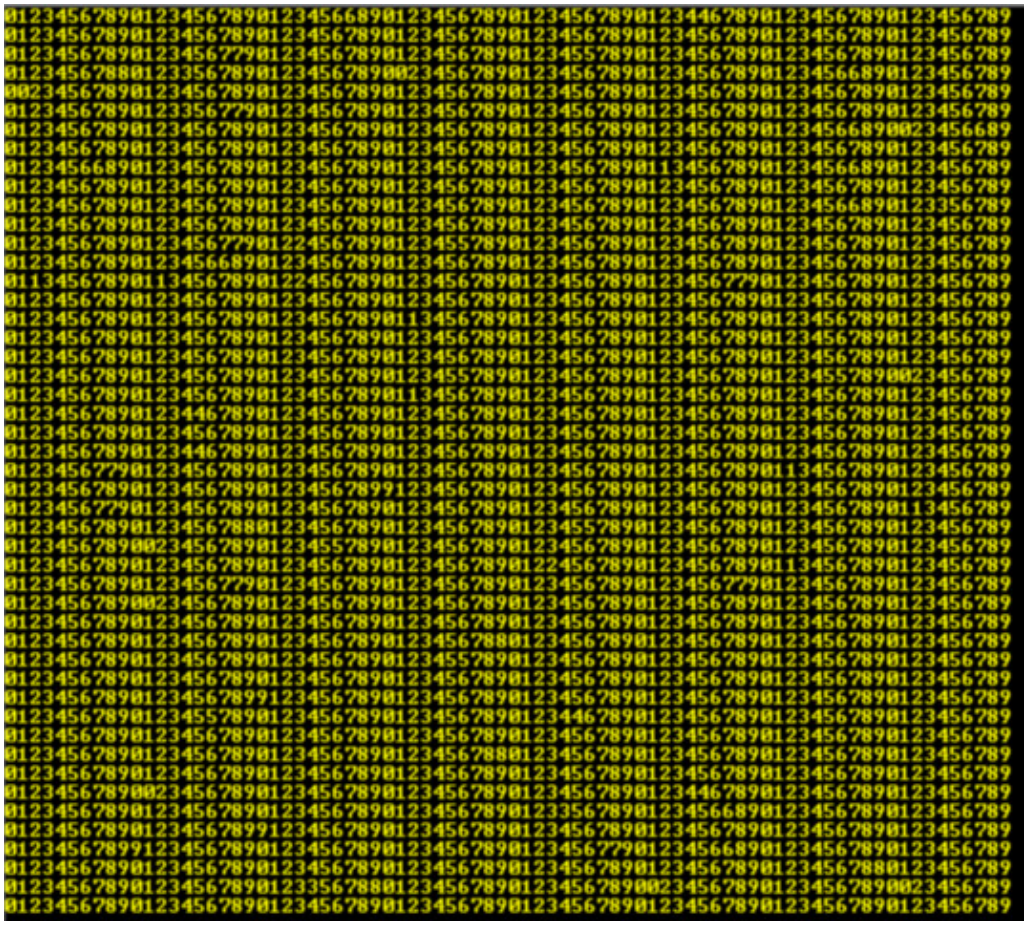
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
012345678901234567890123456789012345678901234567890123**0123**4567890123456789012345678901234567

0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

Here, we see an occasional duplicate DWORD. If we output a single byte at a time rather than 10, the issue occurs mode frequently.



If I add some debug output in the USB library where the data is being put onto the USB FIFO from our buffer pointers I see it being written correctly into the EP FIFO registers in order ok, but different data is being received.

Any hints as to what we're doing wrong here?

Oh yes, our functions for sending are:

```
static int8_t CDC_Init_FS(void)
{
  /* USER CODE BEGIN 3 */
  /* Set Application Buffers */
  USBD_CDC_SetTxBuffer(&hUsbDeviceFS, UserTxBufferFS, 0);
  USBD_CDC_SetRxBuffer(&hUsbDeviceFS, UserRxBufferFS);
  FIFO_Init(&usbRxFifo, usbRxBuffer, USB_RX_BUFF_SIZE);
  FIFO_Init(&usbTxFifo, usbTxBuffer, USB_TX_BUFF_SIZE);
  usbLastTxLen = 0;
  UplinkDecoderInit(&usbRxState);
  state = 0;
  return (USBD_OK);
  /* USER CODE END 3 */
}
uint8_t CDC_Transmit_FS(uint8_t* Buf, uint16_t Len)
{
  uint8_t result = USBD_OK;
  /* USER CODE BEGIN 7 */
  USBD_CDC_HandleTypeDef *hcdc = (USBD_CDC_HandleTypeDef*)hUsbDeviceFS.pClassData;
  if (hcdc->TxState != 0){
    return USBD_BUSY;
  }
  USBD_CDC_SetTxBuffer(&hUsbDeviceFS, Buf, Len);
  result = USBD_CDC_TransmitPacket(&hUsbDeviceFS);
  /* USER CODE END 7 */
  return result;
}
```

Thanks, Dave