**USB Lab**

**Link to video:** http://screencast.com/t/zUbHP9gIBfB

**Task 1: Original code with comments.**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// usb\_bulk\_structs.c - Data structures defining this bulk USB device.

//

// Copyright (c) 2012-2015 Texas Instruments Incorporated. All rights reserved.

// Software License Agreement

//

// Texas Instruments (TI) is supplying this software for use solely and

// exclusively on TI's microcontroller products. The software is owned by

// TI and/or its suppliers, and is protected under applicable copyright

// laws. You may not combine this software with "viral" open-source

// software in order to form a larger program.

//

// THIS SOFTWARE IS PROVIDED "AS IS" AND WITH ALL FAULTS.

// NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT

// NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

// A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. TI SHALL NOT, UNDER ANY

// CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL

// DAMAGES, FOR ANY REASON WHATSOEVER.

//

// This is part of revision 2.1.1.71 of the EK-TM4C123GXL Firmware Package.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_types.h"

**#include** "driverlib/usb.h"

**#include** "usblib/usblib.h"

**#include** "usblib/usb-ids.h"

**#include** "usblib/device/usbdevice.h"

**#include** "usblib/device/usbdbulk.h"

**#include** "usb\_bulk\_structs.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// The languages supported by this device.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**const** uint8\_t g\_pui8LangDescriptor[] =

{

4,

USB\_DTYPE\_STRING,

USBShort(USB\_LANG\_EN\_US)

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// The manufacturer string.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**const** uint8\_t g\_pui8ManufacturerString[] =

{

(17 + 1) \* 2,

USB\_DTYPE\_STRING,

'T', 0, 'e', 0, 'x', 0, 'a', 0, 's', 0, ' ', 0, 'I', 0, 'n', 0, 's', 0,

't', 0, 'r', 0, 'u', 0, 'm', 0, 'e', 0, 'n', 0, 't', 0, 's', 0,

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// The product string.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**const** uint8\_t g\_pui8ProductString[] =

{

(19 + 1) \* 2,

USB\_DTYPE\_STRING,

'G', 0, 'e', 0, 'n', 0, 'e', 0, 'r', 0, 'i', 0, 'c', 0, ' ', 0, 'B', 0,

'u', 0, 'l', 0, 'k', 0, ' ', 0, 'D', 0, 'e', 0, 'v', 0, 'i', 0, 'c', 0,

'e', 0

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// The serial number string.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**const** uint8\_t g\_pui8SerialNumberString[] =

{

(8 + 1) \* 2,

USB\_DTYPE\_STRING,

'1', 0, '2', 0, '3', 0, '4', 0, '5', 0, '6', 0, '7', 0, '8', 0

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// The data interface description string.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**const** uint8\_t g\_pui8DataInterfaceString[] =

{

(19 + 1) \* 2,

USB\_DTYPE\_STRING,

'B', 0, 'u', 0, 'l', 0, 'k', 0, ' ', 0, 'D', 0, 'a', 0, 't', 0,

'a', 0, ' ', 0, 'I', 0, 'n', 0, 't', 0, 'e', 0, 'r', 0, 'f', 0,

'a', 0, 'c', 0, 'e', 0

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// The configuration description string.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**const** uint8\_t g\_pui8ConfigString[] =

{

(23 + 1) \* 2,

USB\_DTYPE\_STRING,

'B', 0, 'u', 0, 'l', 0, 'k', 0, ' ', 0, 'D', 0, 'a', 0, 't', 0,

'a', 0, ' ', 0, 'C', 0, 'o', 0, 'n', 0, 'f', 0, 'i', 0, 'g', 0,

'u', 0, 'r', 0, 'a', 0, 't', 0, 'i', 0, 'o', 0, 'n', 0

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// The descriptor string table.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**const** uint8\_t \* **const** g\_ppui8StringDescriptors[] =

{

g\_pui8LangDescriptor,

g\_pui8ManufacturerString,

g\_pui8ProductString,

g\_pui8SerialNumberString,

g\_pui8DataInterfaceString,

g\_pui8ConfigString

};

**#define** NUM\_STRING\_DESCRIPTORS (**sizeof**(g\_ppui8StringDescriptors) / \

**sizeof**(uint8\_t \*))

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// The bulk device initialization and customization structures. In this case,

// we are using USBBuffers between the bulk device class driver and the

// application code. The function pointers and callback data values are set

// to insert a buffer in each of the data channels, transmit and receive.

//

// With the buffer in place, the bulk channel callback is set to the relevant

// channel function and the callback data is set to point to the channel

// instance data. The buffer, in turn, has its callback set to the application

// function and the callback data set to our bulk instance structure.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

tUSBDBulkDevice g\_sBulkDevice =

{

USB\_VID\_TI\_1CBE,

USB\_PID\_BULK,

500,

USB\_CONF\_ATTR\_SELF\_PWR,

USBBufferEventCallback,

(**void** \*)&g\_sRxBuffer,

USBBufferEventCallback,

(**void** \*)&g\_sTxBuffer,

g\_ppui8StringDescriptors,

NUM\_STRING\_DESCRIPTORS

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// Receive buffer (from the USB perspective).

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

uint8\_t g\_pui8USBRxBuffer[BULK\_BUFFER\_SIZE];

tUSBBuffer g\_sRxBuffer =

{

false, // This is a receive buffer.

**RxHandler**, // pfnCallback

(**void** \*)&g\_sBulkDevice, // Callback data is our device pointer.

USBDBulkPacketRead, // pfnTransfer

USBDBulkRxPacketAvailable, // pfnAvailable

(**void** \*)&g\_sBulkDevice, // pvHandle

g\_pui8USBRxBuffer, // pi8Buffer

BULK\_BUFFER\_SIZE, // ui32BufferSize

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// Transmit buffer (from the USB perspective).

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

uint8\_t g\_pui8USBTxBuffer[BULK\_BUFFER\_SIZE];

tUSBBuffer g\_sTxBuffer =

{

true, // This is a transmit buffer.

**TxHandler**, // pfnCallback

(**void** \*)&g\_sBulkDevice, // Callback data is our device pointer.

USBDBulkPacketWrite, // pfnTransfer

USBDBulkTxPacketAvailable, // pfnAvailable

(**void** \*)&g\_sBulkDevice, // pvHandle

g\_pui8USBTxBuffer, // pi8Buffer

BULK\_BUFFER\_SIZE, // ui32BufferSize

};