USB HID Interface and Host Application Design for the Tiva C Series LaunchPad Evaluation Board

Jennifer Tsang
State University of New York at Buffalo
jtsang3@buffalo.edu
Advised by Dr. Kris Schindler
State University of New York at Buffalo
kds@buffalo.edu

***Abstract*—USB human interface devices (HID) can be used to provide user input to embedded systems applications. As a result, microcontrollers may need to be able to interface with different HID device types. The purpose of this project was to explore the process of configuring a microcontroller for USB host connectivity, implementing interfaces for USB HID devices, and developing USB host applications for embedded systems.**

Using the TivaWare USB library, a host gamepad interface was developed for an ARM Cortex-M4-based microcontroller, allowing the microcontroller to enumerate a Logitech F710 wireless gamepad and receive and parse HID report data from the gamepad. This gamepad interface and a HID mouse device interface provided through the TivaWare USB library were then used to develop applications for controlling an RC car built with the ARM Cortex-M4 microcontroller.

The results of this project can serve as a framework for developing interfaces for other USB HID devices. This report outlines the steps taken to configure an ARM Cortex-M4-based microcontroller for USB host functionality and to develop a host device interface for a USB HID gamepad.

I. Introduction

Human Interface Devices (HID) are peripheral devices that allow humans to interact with computers. HID devices can be used to provide user input to embedded systems. However, not all microcontrollers provide native USB drivers or USB host interfaces for many HID device types.

One example of this is the Tiva C Series LaunchPad Evaluation Board, an evaluation kit for ARM Cortex-M4-based microcontrollers. The TM4C123GH6PM microcontroller in the LaunchPad includes a USB controller, offering USB device, host, and OTG capability. Additionally, the TM4C123GH6PM microcontroller is compatible with the Texas Instruments TivaWare USB library, a collection of C data types and functions for developing USB applications. However, the TivaWare USB library only provides interfaces for USB HID mouse and keyboard devices. In order to interface other HID device types with the TM4C123GH6PM microcontroller, new device interfaces must be implemented.

This project details the configuration of the Tiva C Series LaunchPad for USB host functionality, development of a host interface for a USB HID gamepad, and the utilization of HID device interfaces to develop USB host applications for an embedded system.

II. Gamepad Interface Development

A. Analysis of Gamepad Enumeration and Report Structure

The gamepad utilized in this project was the Logitech F710 wireless gamepad, used in DirectInput mode. Wireshark and the USBPcap USB sniffer tool were used to examine the enumeration process of the gamepad on a laptop and to analyze the report descriptors of the gamepad in order to gain a better understanding of the gamepad's HID report structure.

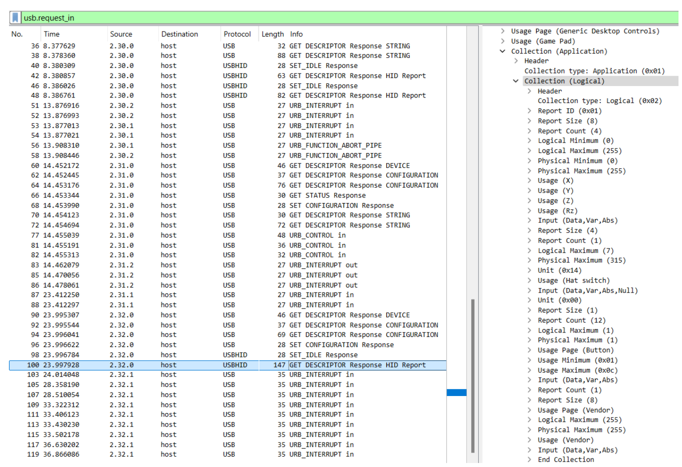

Fig. 1. The HID report descriptor of the gamepad shown in Wireshark.

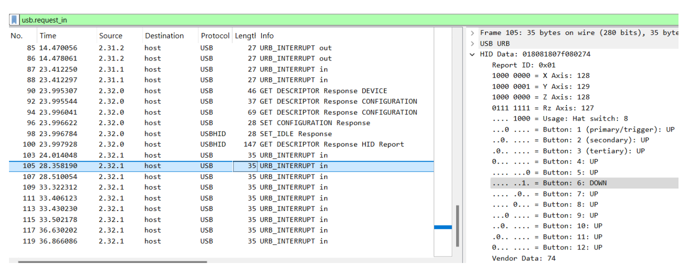

Fig. 2. Example of a gamepad HID report. Both joysticks are in neutral position and button 6 on the gamepad is being pressed.

Based on the HID report descriptor, the gamepad's HID reports encoded three input types - joysticks, a hat switch, and buttons.

The two joystick inputs were each represented as 2 axes, with the gamepad providing 4 total input axes (X, Y, Z, Rz). Each axis was represented by one byte in the HID report with values ranging from 0 to 255 corresponding to how far the joystick was pushed in each direction.

Additionally, the 12 buttons on the gamepad were represented by one bit, which was a 0 or a 1 depending on whether or not the button was being pressed.

Inputs from the hat switch were represented in the report by 4 bits. However, inputs from the hat switch were not parsed or utilized in this project.

B. Hardware Modification

As part of USB host operation, the LaunchPad needed to supply power to any connected USB devices. This was accomplished by soldering a jumper to the H18 test point on the board. The H18 test point connects to USB_VBUS, the power line for the USB connector. By connecting this jumper to an external 5V power source, connected USB devices could be powered from the LaunchPad's USB connector. This connection was also used to power the internal LaunchPad circuit by putting the Power Select switch in the Device position to select USB_VBUS as the board's power source.

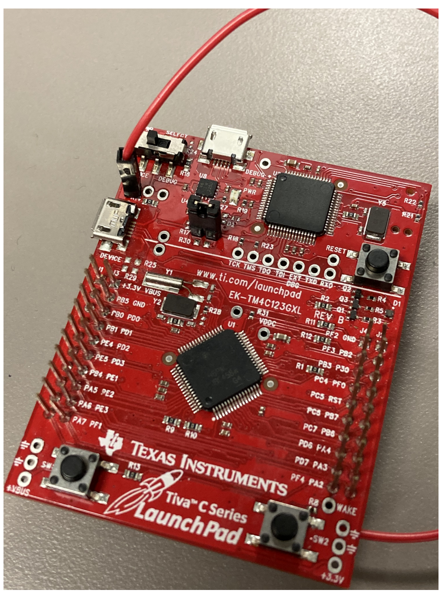

Fig. 3. The LaunchPad with a jumper soldered to the H18 test point.

C. Host Gamepad Interface Software Implementation

To implement the host gamepad interface, functions provided through the host controller driver and HID class driver layers of the TivaWare USB library were used for lower-level details of USB configuration and communication, such as initializing the USB host controller and retrieving descriptor information and HID reports from the gamepad. When new data from the gamepad is available, the host interface reads in the HID report data and parses the report to determine if any button or joystick inputs have changed. Any inputs that have changed are passed up to the USB application layer via an event-driven callback for application-specific processing.

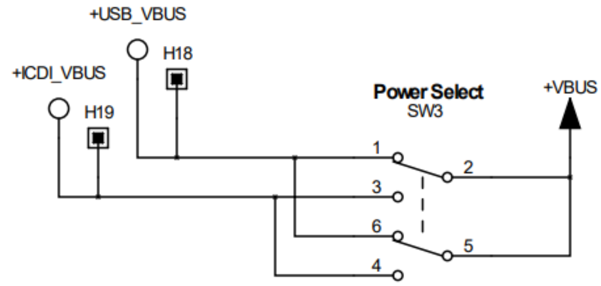

Fig. 4. Schematic for the LaunchPad power source.

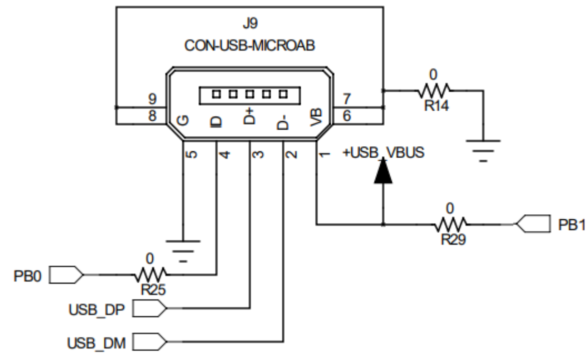

Fig. 5. Schematic for the LaunchPad USB connector.

III. RC Car Embedded System With HID Device Control

Two applications were developed for the LaunchPad to drive an RC car using HID devices. One application allowed for user input from a HID mouse, and was developed using the HID mouse device interface provided through the TivaWare USB library. The second application used the previously developed gamepad interface to take user input from the Logitech F710 gamepad.

A. System Overview

The car was built using a 7.2V NiMH battery pack, 2 DC gearbox motors, and an L298N motor driver for motor direction and speed control. To generate control signals for the motor driver, GPIO and PWM drivers for the TM4C123GH6PM microcontroller were implemented using ARM assembly.

Both applications begin by initializing the necessary hardware peripherals and the USB host stack, as well as opening the desired device interface.

Device connections and removals are handled by the TivaWare library's USB Events Driver, which allows an application to receive non-device class specific events. When a valid HID device is connected or removed, an application-level callback function is called. This function updates the application state so that device initialization and application reconfiguration can occur as needed.

When input is received from a HID device, the device interface parses the HID report to extract the updated joystick

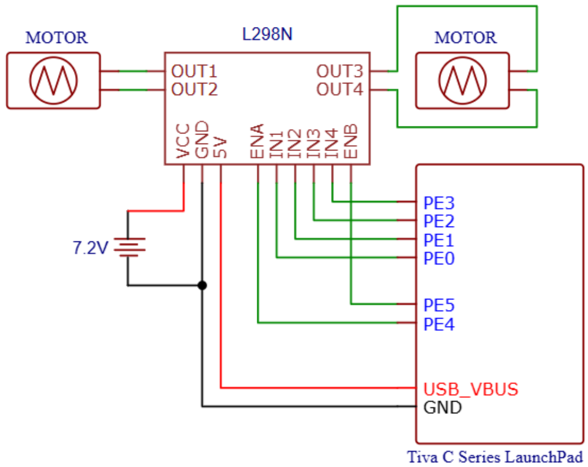

Fig. 6. Schematic for the RC car.

and button states. The updated input states are passed to the application, where they are saved for use in the main program.

B. Mouse-based Control

The left mouse button, right mouse button, and middle mouse button are used to drive the car. Pressing the middle mouse button toggles the car's direction between forward and backward. The left and right mouse buttons individually control the left and right motors, respectively. Pressing these buttons drives the corresponding motor at full speed.

C. Gamepad-based Control

Two gamepad-based control schemes for the car were implemented.

First, the bumper and trigger buttons can be used to drive the car at full speed. The left trigger and bumper drive the left motor forward and backward, respectively. The right trigger and bumper control the right motor similarly.

Alternatively, the joysticks can be used for analog control of the motors. Pushing the left joystick forward or backward drives the left motor forward or backward, respectively, with motor speed changing proportionally to the displacement of the joystick. The right joystick controls the right motor in a similar fashion.

IV. Conclusion

The LaunchPad was successfully configured for USB host functionality and a host device interface for a USB HID gamepad was developed. This interface was then used in the implementation of an RC car embedded system, allowing for gamepad-based user input. The steps taken to configure the LaunchPad and develop a gamepad device interface can serve as a framework for development of other USB HID device interfaces, so that more HID device types can be used in embedded applications utilizing the TM4C123GH6PM microcontroller.

V. Future Work

Some suggestions for future work include:

- Implementation of USB host interfaces for other USB HID device types.
- Development of a USB host application supporting multiple device types. Currently, only one device interface can be opened at a time within an application using the TivaWare library, so applications cannot interface with multiple device types. Addressing this may involve modifying the TivaWare library to add multi-interface support.
- Configuration of the LaunchPad for USB device or OTG functionality. For USB OTG configuration, additional hardware modifications must be made in order to allow the USB controller to control the USB ID and USB VBUS signals of the USB connector.

References

[1] Texas Instruments. *Tiva C Series TM4C123G LaunchPad Evaluation Kit User's Manual*. (2013). [Online]. Available: https://www.ti.com/lit/ug/spmu296/spmu296.pdf?ts=1734889760629

[2] Texas Instruments. *TivaWare™ USB Library for C Series User's Guide (Rev. E)*. (2020). [Online]. Available: https://www.ti.com/lit/ug/spmu297e/spmu297e.pdf?ts=1734883024814

[3] Texas Instruments. *Tiva™ C Series TM4C123GH6PM Microcontroller Data Sheet (Rev. E)*. (2014). [Online]. Available: https://www.ti.com/lit/ds/spms376e/spms376e.pdf?ts=1734923071071&ref_url=https%253%252F%252Fwww.ti.com%252Ftool%252FEK-TM4C123GXL

[4] C. Tsai, "Using USB Host Mode on the EK-TM4C123GXL LaunchPad." ti.com. https://www.ti.com/lit/an/spna243/spna243.pdf?ts=1734923106614&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FEK-TM4C123GXL

[5] "USB Human Interface Devices." OSDev.org. https://wiki.osdev.org/USB_Human_Interface_Devices

[6] "Introduction to HID report descriptors." docs.kernel.org. https://docs.kernel.org/hid/hidintro.html