USB Keyboard mouse using PDIUSBD11

Application Notes

Introduction

The PDIUSBD11 (D11) is a high speed USB interface device. It can be interfaced to a micro-controller using a minimum of 3 I/O lines, a pair of I^2C lines and an interrupt line. It offers 2 configuration, either a single device or three embedded devices.

There is a total of 4 endpoints for a single device configuration. Endpoint 0, the default Control endpoint and endpoint 1-3 are generic, which can be used as either interrupt, bulk or control endpoints. Each endpoints have a buffer size of 8 bytes.

In the 3 embedded function configuration, Embedded function 1 consists of 2 endpoints, a default Control endpoint 0 and generic endpoint 1. Similarly, for embedded function 2-3.

The D11 can be used as a HID device (though not restricted to a HID), which, this application notes shows how a keyboard with PS2 mouse can be implemented.

Hardware description

This firmware uses the hardware configuration of the Philips keyboard hub. As the PCB is a general release for a keyboard hub, the use of it just as a keyboard with PS2 mouse does not require the use of the downstream ports. The ports can be safely ignored because they do not contribute nor affect the performance of the USB Keyboard cum PS2 mouse.

Firmware description

Files description

The firmware implements software I^2C to control the D11. All the routines are contained in the main kbms1_7.asm file. Along with the assembly language file, a file named **transctn.typ** defines the transaction codes used to passed between the Control OUT endpoint routine to the Control IN endpoint routine. In another file, **H11.cmd**, contains the I2C commands of the D11, it is a subset of the H11/H11A commands, functionally equivalent.

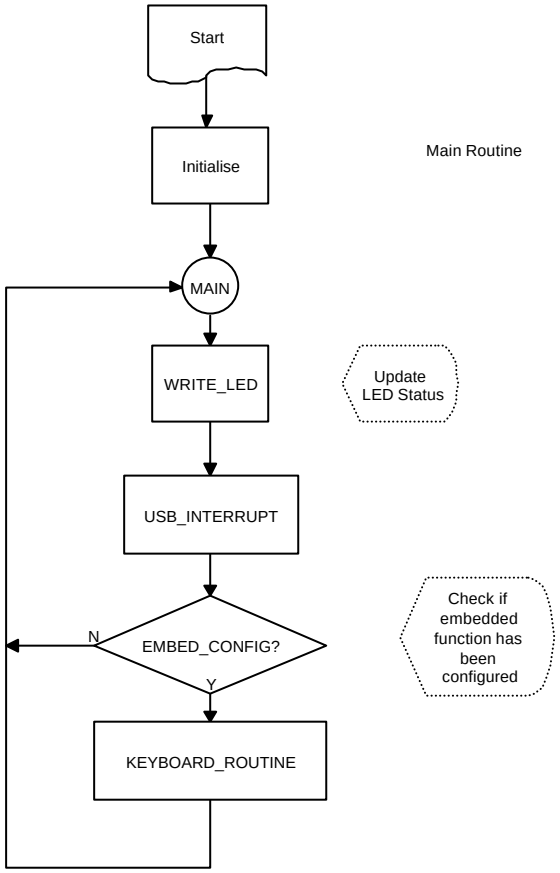
USB Class description

The firmware implements a HID compliant composite device. It is a USB device with a single configuration but two different interfaces. Each interface defines it's own interrupt endpoint and report descriptor. Interface 0 defines a USB keyboard with endpoint 1 used as the interrupt endpoint for key-pressed data while Interface 1 defines a USB mouse using endpoint 2 as the interrupt endpoint for tracking the movements of the mouse.

Flow Chart

The main loop polls for USB interrupt from the D11 as well as running a routine to detect key-press events and mouse movement.

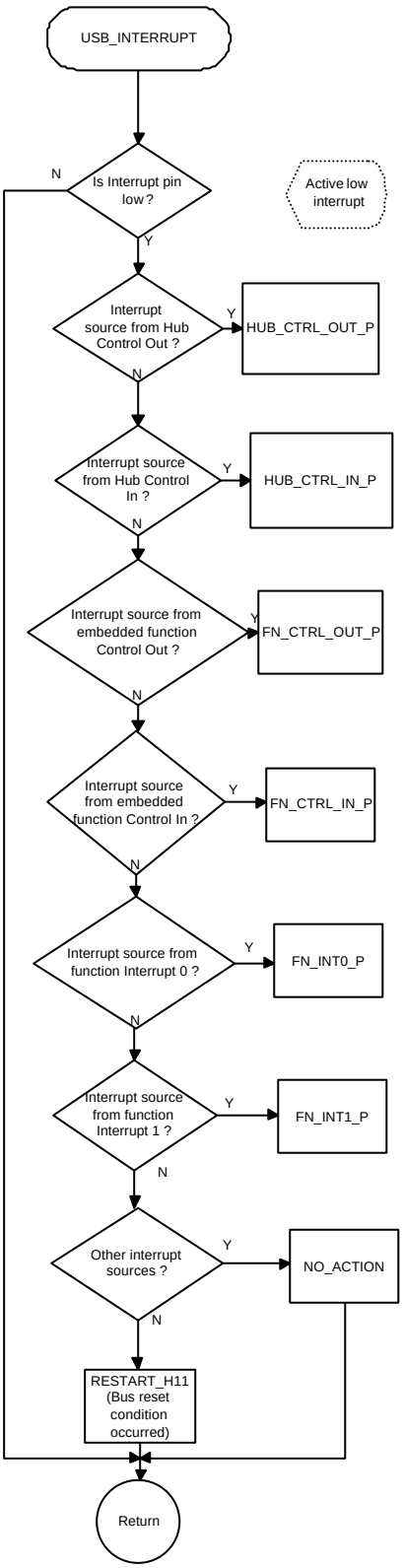
a) Main loop



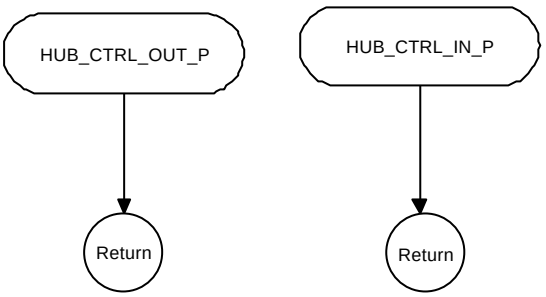
The D11 firmware can be used on a H11A because both are software compatible. Hence, the main interrupt routine points to a Hub interrupt as well. However, the interrupt is never called.

Under the condition that the interrupt pin was activated without a source from byte 1 of the Interrupt register, this indicates that a BUS reset condition has occurred as endpoint index 8 and 9 is not used in the current configuration. Bus-resets can also be determined by reading the 6th bit of the 2nd byte of the Interrupt register.

b) USB Interrupt

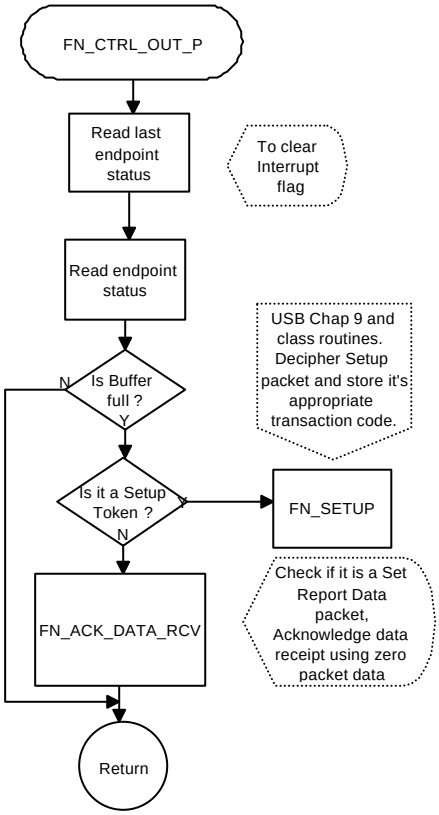


c) Hub Control IN and OUT routine

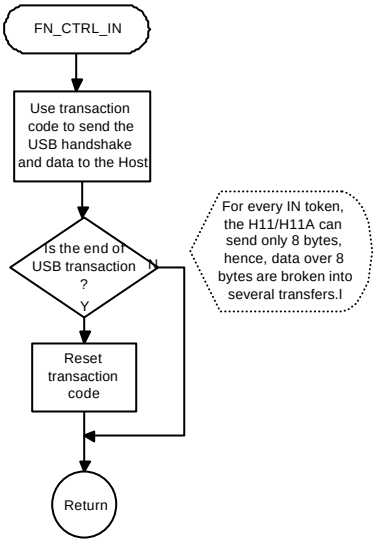


d) Function Control OUT routine

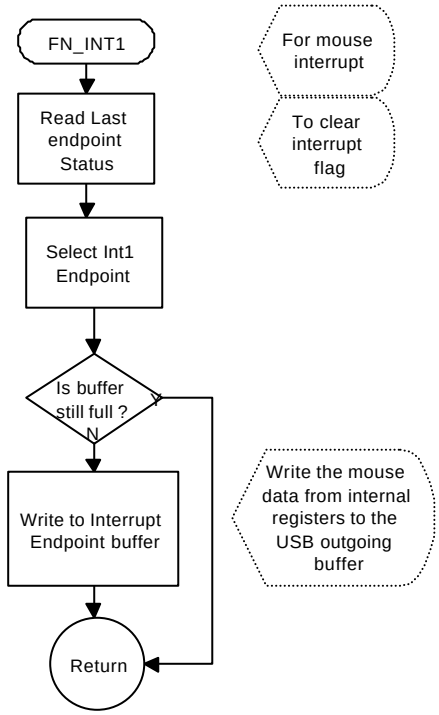
The deciphered SETUP token is kept as a transaction code described in the file **transctn.typ**. The actual data transfer is performed by the Function Control IN routine.



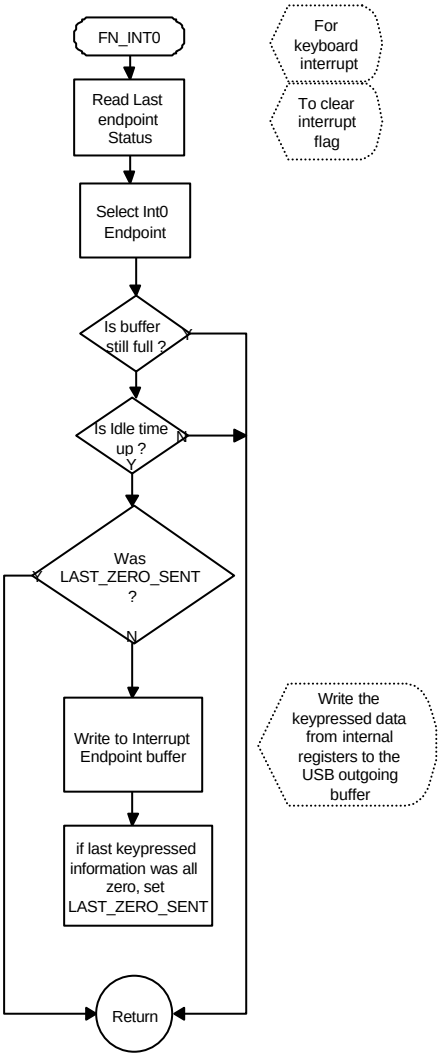
e) Function Control IN



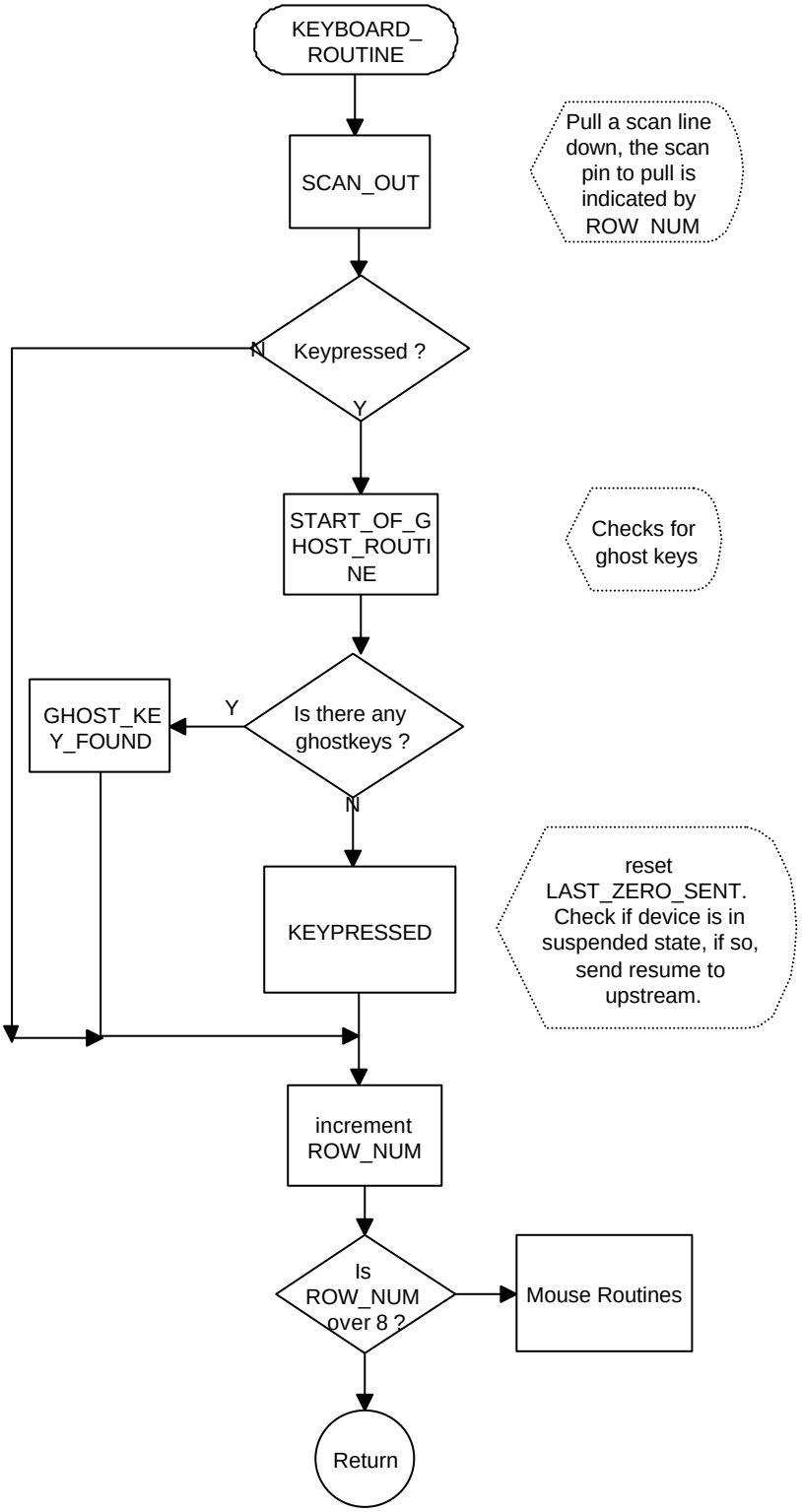
f) Function Interrupt 1. The mouse interrupt routine is called for each IN token for endpoint 2.



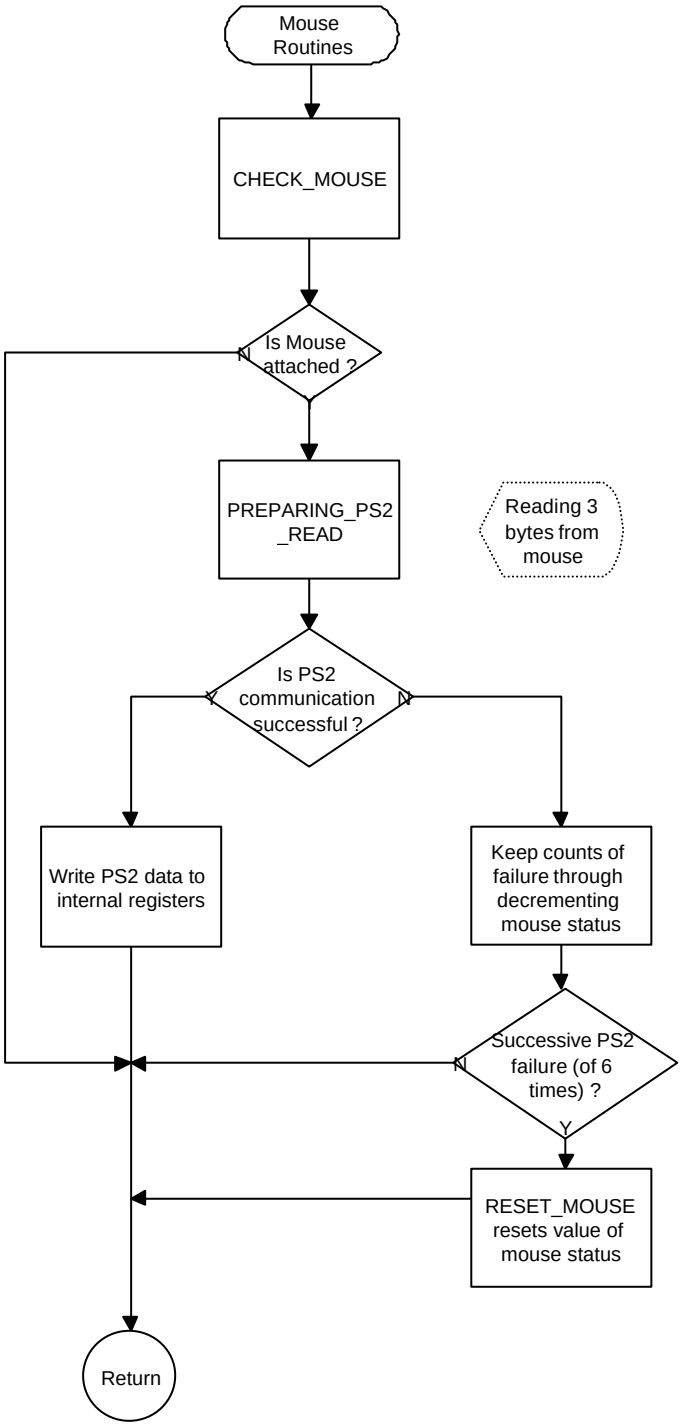
g) Function interrupt routine. With the Debug Mode on, this is called whenever there is an IN token arriving on the Endpoint described by the endpoint descriptor. In this case, this is the keyboard interrupt IN token.



h) The keyboard routine



i) The mouse routine


Mouse Routines
CHECK_MOUSE
Is Mouse attached ?
N
Y
PREPARING_PS2 _READ
Reading 3 bytes from mouse
Is PS2 communication successful ?
Y
N
Write PS2 data to internal registers
Keep counts of failure through decrementing mouse status
Successive PS2 failure (of 6 times) ?
N
Y
RESET_MOUSE resets value of mouse status
Return