

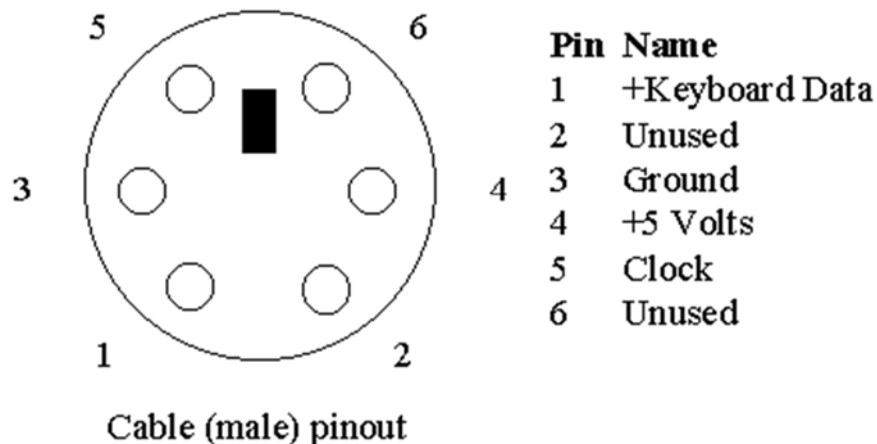
PS/2 Keyboard Tutorial

This tutorial provides an overview on how to use the PS/2 input on the DE1-SoC board.

NOTE: USB-2-PS/2 converters do NOT seem to work for most USB keyboards. If you are going to use a keyboard, either find an old PS/2 keyboard, or an old USB keyboard that comes with its own converter. Do NOT take a random USB keyboard and just toss a converter on it – most will NOT work.

PS/2

PS/2 Keyboard and Mouse Cable



The pinout for the PS/2 Keyboard and Mouse cable is shown above. Through a clock pin and a serial data pin, the FPGA connects to the clock and serial line keyboard data line of the PS/2 cable. The process of process clock signals and serial keyboard data is handled by the provided Verilog files and should not be a concern of the user.

Provided Interface Overview

Two interfaces describe methods to view the keyboard data, without directly concerning the PS/2 clock and serial data lines. One interface, the raw interface, allows direct access to the keyboard data processed after a key is pressed, held down, or released. The other interface is simpler and provides a signal when a key is pressed or released and then a byte to represent the key that is pressed or released.

Representation of Keyboard Data

When a key is pressed, released, or held down, a packet of information known as a “scan code” is sent by the keyboard over the PS/2 data line. Scan codes fall into the category of “make codes” or “break codes”. When a key is pressed or held down, a make code is sent. While a key is pressed down, the make code is continually sent at a certain interval. When a key is released, a break code is sent. Every key has a unique make code and break code.

There are three standard scan code sets, named one, two, and three. The default for modern keyboards is set two.

The scan code sets are included in the zip for this set of drivers and documentation as .html files with tables for the scan codes.

Behavior of Multiple Keys held

If one key is held down, its make code is sent at a periodic rate. If another key is then held down, the make code of the new key held down is sent at a periodic rate instead. If the first key is released, the break code for the first key is sent once and the make code of the second key continues to stream at a periodic rate. If the second key is released, the break code for the second key is sent and no new scan codes are sent until a new key is pressed or the first key is released.

The number of keys that can be held down following this behavior depends on the circuitry of the keyboard used. It can be assumed that 2 keys can be held down for any keyboard and for any 2 keys. If too many keys are held down, then the scan code 0x00 is sent at a periodic rate until the keys over the maximum are released.

Interface for Simple Press Behavior

NOTE: This interface only works with keyboards that use scan code set 2 or 3, which encompasses most modern keyboards

keyboard_press_driver.v describes a module that returns keyboard data when a key is either pressed or released.

When a key is pressed, a valid signal output goes high and a makeBreak bit is set to show where a make or break was sent. Makes are key presses and breaks are key releases. The outCode byte register is set with the keyboard data. This byte is the scan code for the given key. Note that a break code is usually 2-3 bytes, but only the unique last byte is given because it is enough to identify the key and it is almost always the same as the one byte make code.

- User Inputs
 - CLOCK_50 – 50 MHz clock
 - reset – active high reset signal
- User Outputs
 - valid – high when a make or break code is ready to be read
 - makeBreak – 1 for make and 0 for break
 - [7:0] outCode – a byte representing keyboard data, (take note of this one byte representation from the description above)
- Top Level Inputs
 - PS2_DAT – ps/2 data line
 - PS2_CLK – ps/2 clock

Users should connect the inputs and outputs to their design and the top level inputs should connect to the top level pins of the same name.

Interface for Raw Scan Code Driver

keyboard_scancoderaw_driver.v describes a module that returns keyboard data one byte at a time when a key is pressed, held, or released.

When a scan code byte is ready, a scan_ready signal goes high and a scan_code byte register is loaded with the given byte. Scan code bytes are sent when a key is pressed or released, or at a periodic rate when a key is held. Note that this driver shows each byte at a time. Some make codes and most break codes are more than one byte.

- User Inputs
 - CLOCK_50 – 50 MHz clock
 - reset – active high reset signal
- User Outputs
 - scan_ready – high when a new scan code byte is ready to be read
 - [7:0] scan_code – a byte representing part of a scan code
- Top Level Inputs
 - PS2_DAT – ps/2 data line
 - PS2_CLK – ps/2 clock

Users should connect the inputs and outputs to their design and the top level inputs should connect to the top level pins of the same name.

Tutorial and code developed by Kyle Gagner and Jesse Liston with Professor Scott Hauck

keyboard_inner_driver.v and oneshot.v are adapter from/provided by Dr. John S. Loomis, Professor Emeritus of Electrical and Computer Engineering at the University of Dayton.

<http://www.johnloomis.org/>