School of
Advanced Technology

CST8227 Lab 2: Keyboard and Mouse Emulation

Due date: the beginning of your lab period, Week 3

What you will do:

1. Confirm proper operation of the Teensy (and associated drivers!) for keyboard & mouse
2. Learn a few new keystroke combinations to control MS Windows (or Linux if you prefer)
3. Customize Windows by adding shortcut keys for a number of applications
4. Create a customized app which uses the Teensy keyboard and mouse modes
5. Experiment with other interesting ways of using a Teensy

What you need to submit and when:

1. Complete the pre-lab on Blackboard for this lab.
2. Complete a demo of [2 marks each item]:
    (1) the simple app which records text in a Notepad file;
    (2) your mouse app which draws geometric shapes (eg. in Paint);
    (3) your third application using keyboard and/or mouse control.
3. The online post-lab quiz.

Required Equipment:

- The Arduino IDE installed and verified functional
- A working network connection.
- A Teensy 3.1 with pins, mounted into the protoboard.

Marks:

Each of the demos and postlab identified above are weighted equally, and combined to generate your mark for this week's lab.

References and Resources:

- Teensy keyboard tutorial: http://www.pjrc.com/teensy/td_keyboard.html
- Teensy mouse tutorial: http://www.pjrc.com/teensy/td_mouse.html
- Teensy serial port tutorial: http://www.pjrc.com/teensy/td_serial.html
- Any other URLs, as given in this lab
- An Arduino compatible “Teensy”

Task 1: Learn and understand Teensy for keyboard and mouse operation

The Teensy has an advantage over the original (genuine) Arduino in that is provides keyboard and mouse emulation out-of-the-box. There are less than a dozen calls in the API, so it's easy and useful to explore them all. The serial mode is also extremely; you'll find it especially handy for debug messages since the Arduino has no direct way of displaying text output.

Step 1**:** Run through the USB keyboard tutorial: www.pjrc.com/teensy/td_keyboard.html
Confirm that everything works as advertised.
Step 2: Run through the USB mouse tutorial: www.pjrc.com/teensy/td_mouse.html
Confirm that everything works as advertised.
Step 3: Run through the serial port tutorial: www.pjrc.com/teensy/td_serial.html
Confirm that you can successfully see the serial port messages.

Task 2: Find keyboard shortcuts for Windows

The downside of GUIs is that the position of items can easily change. Keyboard shortcuts are good because they “always” work. In a programmatic environment, using keyboard shortcuts is preferable to mouse events so we'd like to know as many as possible.

Step 1: Google to find the best, most comprehensive set of keyboard shortcuts for your version of Windows (or Linux if you prefer).
Step 2: Try out the shortcuts to confirm that they work.

Task 3: Customize Windows with keyboard shortcuts

MS-Windows has long had the ability to designate a particular keystroke combination to activate a particular application. Make sure you know how this works.

Step 1: Start simple: add a shortcut keystroke for a single application. Right-click on an application launcher (either on the desktop or within the Start menu); choose Properties, and look for the setting Shortcut Key
Step 2: Experiment to find out the requirements for a keyboard shortcut: does it always require a CTRL key? a SHIFT key? both these keys?
Step 3: Find out if it's possible to (accidentally) assign the same keyboard shortcut to two different applications.
Step 4: Explore the behaviour of issuing the keyboard shortcut multiple times, putting the application in the background between keystrokes. Do you get multiple instances of the application, or does it just get brought to the foreground?
Step 5: Explore any other behaviour you can imagine for keyboard shortcuts.

Task 4: Explore mouse events

Use a procedure similar to the previous task to explore the operation of mouse events. Make sure you understand the behaviour you'll get when using mouse events programmatically.

Task 5: Create a Teensy keyboard (and mouse) app

- Create a simple app to demonstrate the operation of the Teensy keyboard (and possibly also the mouse). For example, the app opens notepad and types some initial text, adds additional text every 30 secs and ensures the file is saved to disk **every time** text is added.
- Create an app to demonstrate the operation of the Teensy mouse. For example, the app opens Paint (or any equivalent program) and draws a series of geometric shapes (square, triangle, circle). As a minimum, your app **must draw two different shapes**, and **must draw a circle** for full marks.

Task 6: Create a Better Teensy keyboard/mouse app

Here's a chance to both have fun and impress the Professor. Create an app that does something more useful. For example, can you:

- Programmatically invoke an email app, start and fill in an email, and send it off (eg. once every minute)?
- Invoke a browser, navigate to a suitable web page, and send an SMS message at a given interval?
- Do anything else that you can (convincingly!!) support as a useful app, such as automating a tedious and error prone keyboard/mouse sequence?