Keyboard Hacking (Part I)

Warning!

- Keyboard hacking can prove hazardous to the health of your computer.
- There is a +5 voltage presence on the keyboard encoder while connected to the computer.
- If something goes wrong, it is possible to fry the keyboard port on the computer or the motherboard itself.
- Follow the step-by-step guidelines can avoid unexpected result.

Don’t:

- Don’t connect any external power supply to the keyboard.
- Don’t plug/unplug any hardware in the machine without permission.
- Don't plug/unplug keyboard (or non-USB device) inside SCM LAB machines after the workshop, unless you get the permission from Ryan (in SCM lab you want to test your assignment/project).
- Don’t leave any rubbish in SCM lab (e.g. screws, screwdriver, wires, switches, circuit, electrical tape, rubber pads, keyboard case……etc).

Do:

- Ask question if you have any problems.
- Keep the SCM lab clean and tidy during/after the workshop.

Created By Ryan Lam
How Computer Keyboards Work?

A keyboard allows you to press a key, which closes a circuit associated with a particular keystroke. That closed circuit is detected by a microprocessor inside the keyboard called a **keyboard encoder**, which recognizes which key has been pressed. The keyboard encoder takes that information, encodes it in a digital form the main computer can understand, and passes it to the computer via the keyboard port.

Inside the keyboard:

1. keyboard encoder
2. rubber pads
3. **Key matrix** (or called Mylar sheet)
Keyboard Technologies

- The **key matrix** is the grid of circuits underneath the keys.
- The circuit is broken at the point below a specific key.
- Pressing the **rubber pad** bridges the gap between 2 halves of a circuit, allowing a tiny amount of current to flow through.
- The **keyboard encoder** monitors the key matrix for signs of continuity at any point on the grid.
- When it finds a circuit that is closed, it compares the location of that circuit on the key matrix to the character map in its ROM.
- The character map is basically a comparison chart for the processor that tells it what the key at x,y coordinates in the key matrix represents.

![An example of a possible matrix using the keyboard encoder](image)

**Note:** A matrix is a method of using a small number of contacts to account for a larger number of inputs by arranging them into a grid. (e.g 26 contacts → 18 rows x 8 columns = 144 combinations)
Keyboard hack advantages:

1. **Low Cost** – $15 for a keyboard and some basic tools (screwdriver, scissor, crocodile clips, wires, tape, switches, soldering tools, hot-melt gun/glue)
2. **Number of total inputs** – A keyboard hack will allow a total of 101 (or 104) separate inputs, however only probably 16 of these can be pressed simultaneously.
3. **Size** – The circuit cards on normal keyboards measure approximately 1.25 by 3 inches. This may be an important consideration for a desktop controller with many buttons.
4. **Feeling of accomplishment and fun**

Keyboard hack disadvantages:

1. **Time consuming** – Mapping the matrix, planning the input keys, testing, soldering…etc
2. **Number of Independent Inputs** – A keyboard hack will support from 12 to 18 independent inputs (pressed simultaneously)
3. **Ghosting, Masking, and Key Blocking problems**
4. **Safety problems** – Keyboard hacking can prove hazardous to the health of your computer.
5. **Soldering skill is needed**

Tools required:

1. **Cheap Keyboard** -- Only the Integrated Circuit card from the keyboard will actually be used, so don't spend a lot of money on this item. Normal PS/2 keyboard is okay. However, DONT buy a USB keyboard (and do not hook your PS/2 Keyboard up to the USB port) as the standard USB interface only supports 6 simultaneous key-presses.
2. **Basic tools** -- screwdriver, scissor, tape, hot-melt gun/glue, soldering tools (e.g soldering iron, iron stand, de-soldering pump, solder wire)
3. **Wires, crocodile clips and switches** – we need them for testing the key matrix and reconstructing the circuit and interface
4. **Keyboard Testing Software** -- allow you to see what keys are really going to the computer when you press the keyboard keys.
Let’s start to hack your keyboard!

[Step 1] – Disable the “shut down” function in window (by power/sleep buttons)

In window 2000/XP:

2. Select the Advanced tab.
3. In the “When I press the power button on my computer” list, select “Standby” (select “Do nothing” for winXP) and press OK
4. In the “When I press the sleep button on my computer” list, select “Standby” (select “Do nothing” for winXP) and press OK.
[Step 2] – Prepare a screwdriver and keyboard

[Step 3] – Let’s check out the keyboard (cost $15 only…not bad la)
[Step 4] – Remove the screws

Place the unplugged keyboard face down on the desktop and remove all the screws.

There are more then 10 screws :<

Cheer up! You can do it!

Care! Secret screw is under the label

I got it.
Keep the SCM LAB clean and tidy!

[Step 5] – Look inside the keyboard

Remove the back cover **carefully** from the keyboard. Now, you may find a keyboard encoder, 2 Mylar sheets, and plastic pads
[Step 6] – Take out the keyboard encoder

Take out the circuit board carefully.

⚠️ **CAUTION!** There are 4 thin wires (white, green, red and yellow) connected to the circuit board. They can be broken easily.

⚠️ **DONT** try to connect any other power supply, devices or wires to these terminals.
[Step 7] – Testing a key

- Shut Down the computer and unplug the keyboard
- Plug the circuit card into the keyboard port.
- Restart the computer. Open a notepad program
- Prepare a wire with crocodile clips
Attach one end of the wire to one of the contacts. Then, hold the other end of the wire and touch with other contact. Like as follow:

⚠️ CAUTION! Don’t cross with other contacts without clear instruction
■ Close-up

■ Observe the keystroke that is generated on the computer

■ Okay, let’s remove the wire first.
■ Now, you may ask: “How to find the rest of keys?”
■ Answer: you need a **paper**, a **pen** and **Keyboard Testing Software**.
[Step 8] – Mapping the matrix

Now, we are going to the most time-consuming step -- **Mapping the matrix**. Because every keyboard is different, you will have to manually determine the matrix your keyboard uses. In our class, we are using a 18 x 8 matrix keyboard. Let’s start to map the matrix:

- Draw a grid on a piece of paper matching the X and Y contacts on your keyboard encoder.
- Find a program on your computer that will tell you what keystrokes are being generated.
- Record the generated keystrokes to map the matrix.

Define the X and Y contacts on keyboard encoder:

```
<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Draw a grid to match the matrix (X = column, Y = row)
Keyboard Testing Program

It's easy to tell when letters and numbers are being pressed with any word processor or notepad application. However, those won't tell you if you're pressing the left shift key, right control key, etc. Luckily, there are several utilities that will do just that. In our class, we use **KeyScan**.

---

**KeyScan** - When the program starts, a screen appears with a picture of a keyboard and all the keys grayed out.

- When a key is pressed, the key on the keyboard display turns black.
- In addition, a counter on the screen keeps track of how many keys have been pressed and the program beeps if key blocking is encountered until the blocking condition is removed.

---

**Minor bug**: the program cannot distinguish between the **Enter** key and the **Keypad Enter** key and will show both of these as pressed and count 2 key-presses when either one is pressed.
Key Matrix

Now, you can use this page to record all the generated keystrokes to map the matrix.

Good news! Six of them are given to you. See below:

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>Q</td>
<td>Tab</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y2</td>
<td>W</td>
<td>Caps Locks</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If some combinations don’t generate any response in the keyboard testing software, then just leave the box blank. It is normal, because window only supports 104 keys. Therefore, there are around 40 unused inputs for 18 x 8 matrix keyboard.
[Step 9] – Connecting a switch

Now, we may use a switch to connect to the keyboard encoder.

- Use two **crocodile clips** to attach to the terminals (e.g. X3 - Y7 = L key). Like as follow:

- Attach the other ends of the clips to a switch.

- Now, press the switch. It should generate a "L" keystroke!
[Step 10] – Choosing the keys for Flash

After the mapping the matrix, you may plan how many keys are needed and choose the correct keys from the matrix to prevent “Key Ghosting or Blocking” problems. Finally, you can assign those keys to your flash program.

---

Keyboard hack problems:

**Key Ghosting** --- when you press any 3 keys that form a rectangle (any number of rows and columns apart) in the matrix. For example, pressing Q, W, and A, simultaneously will generate S.

![Diagram of key matrix]

Q + W + A = QWAS

Why this can happen? Recall that a keyboard encoder works by detecting completed circuits. When you press "Q," circuit X1-Y1 is completed. While pressing "Q," also press "A" to complete circuits X1-Y1 and X1-Y2. Add "W" and something interesting happens. Not only have you added circuit X2-Y1, but because the three keys "Q," "A," and "W" involve all four terminal points, there are complete circuits between all combinations including X2-Y2, even though no key was physically pressed at X2-Y2 and so generates a phantom "S" keystroke.

**Key Blocking** – In most of the newer keyboards, a technique called key blocking is used. In this method, the IC scans the matrix and blocks any keys which could create ghosting or masking. Thus when any three keys which would form a square if a fourth key is pressed are depressed, the third key will not register. Using the example above, when “Q”, and “A” are both pressed and “W” is then pressed, the “W” key will not register.
Wire connecting:

Now, some of you may find that it is not a good idea to use too much wire with crocodile clips connecting to the keyboard encoder. (Clip’s size is too big and easy to lose). Later, you will learn some basic wiring skill.

References:

http://www.extremetech.com

http://www.mameworld.net

http://www.howstuffworks.com/