Reverse Engineering Project 2004

The disassembly and analysis of a standard computer keyboard

a.k.a.

"This wuz a k3ybored"

Group Members:
Steven Maresca 203-605-9533 steven.maresca@uconn.edu
Justin DeMaris 427-2127 justin.demaris@uconn.edu
Matthew Rice 427-4533 matthew.a.2.rice@uconn.edu


COMPAQ


THIS WUZ
A
K3YBORED

Overview Of Presentation

- Research
  - Layout
  - Keyboard Varieties
  - Mechanics
  - Electrical
- Disassembly
- Probing
- Results

Research

Mechanical Aspects

• Keyboard Layouts and Varieties
• Size and Shape
• Keyboard Components:
- Keycaps
- Key Switches
- Key Pitch
- LED’s

Keyboard Layouts

- Many and varied
- Include:
  - Alphabetic layouts
  - The QWERTY layout
  - The Dvorak layout
  - Other experimental layouts

Alphabetic

- Arranged alphabetically
- A typewriter based on such a layout was granted a patent in 1868, to designers Scholes, Gidden, & Soule
- Developed for “hunt and peck” railway ticket typists as protection against forgery
- Two-row design (A-M & N-Z)

QWERTY

- Three-row design named for the first six letters in the top row (QWERTYUIOP)
- Patented in 1878, by designer Sholes
- Reasons for implementation of QWERTY are many and varied.
- The most common keyboard layout today

Why QWERTY?

- Purportedly designed to help prevent jamming in sluggish, clumsy mechanical typewriters by spacing out common keys (and thereby slowing type rate)
- The design, while not especially efficient, was well supported, well marketed, well known, and most importantly, widely implemented.

Alternative Layout - Dvorak

- Designed by Dr. August Dvorak and William Dealey in the late 1930’s / 1940’s.
- Designed to be more efficient
  - In QWERTY keyboard, 31% of all strikes are in the home row
  - In Dvorak, 70% of strikes are in home row

Dvorak Layout

Dvorak Analysis: Advantages vs. Disadvantages

Advantages:

- After 2 weeks of typing, the user regains the speed of the QWERTY keyboard, and then passes it
- Less work for the hands, meaning long typing at high speeds is less strenuous

Problems:

- Requires a complete relearning of typing because of the new layout (versus the now standard QWERTY layout)
- If the user uses multiple computers, it may be necessary to be able to use both QWERTY and Dvorak, which is difficult

Setting up the Dvorak Layout

- Hardware solution:
  - Hardwired Dvorak keyboards are available online from many retailers
  - Programmable keyboards (also available from many retailers) can be easily programmed to use the Dvorak layout
- Software solution:
  - Tell you're operating system to use a Dvorak layout

Student Experience

All three members of this reverse engineering group have coincidentally attempted to use the alternative Dvorak keyboard layout. Each used a QWERTY keyboard with altered regional settings (a Windows settings adjustment), and in one case a modified QWERTY keyboard. Their experiences found that it was relatively simple to learn the layout and achieve typing speeds at least on par with QWERTY and often in excess. Coinciding with the Dvorak prediction, the students found that many common words such as “the”, “and”, and “said” could be typed exclusively using the home row. It is estimated that with further practice, the Dvorak keyboard could yield results significantly faster than those achievable using QWERTY. Of course, the main obstacle in mastering the Dvorak layout was and remains overcoming the now innate and instinctive QWERTY layout.

Alternative Keyboard Varieties

- Single-Handed Keyboards
- Split Keyboards
- Chord keyboards
- QWERTY Variant Keyboards
- Projection and Sensor Keyboards

Single-Handed **Keyboards**

- Since most OS's are graphical now, the mouse is a major pointing device
- This leaves one hand left for keyboard usage
- Available in many different formats, including Dvorak and QWERTY

MALTRON Single-Handed Keyboard

- Available in right or left-handed versions
- Can be difficult to learn because of the vast number of keys that one hand has to use and a non-standard layout
- Not very portable

http://www.maltron.com

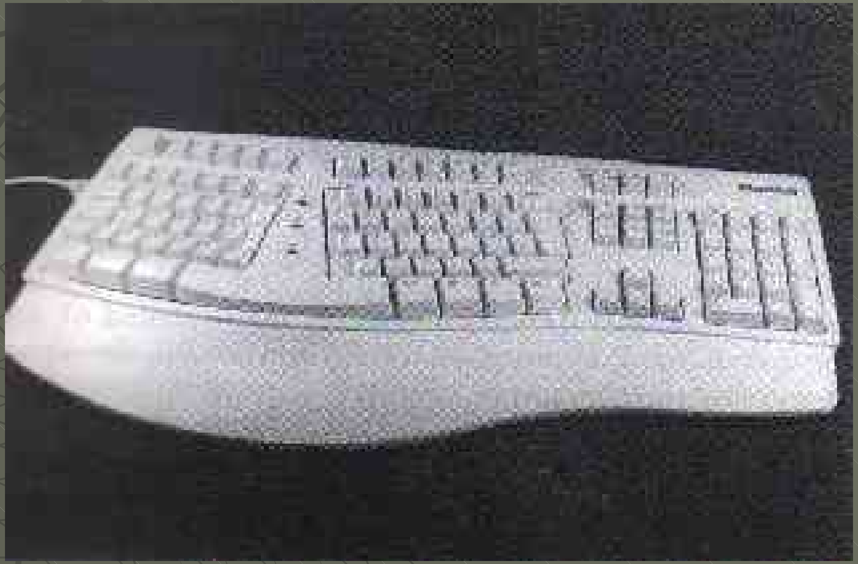
FrogPad Single-Handed Keyboard

- Not very ergonomic
- Completely new layout
- Very portable keyboard, so once you learn it, it can be transported easily to any computer you use

http://www.frogpad.com/

Split Keyboards

- Developed to reduce cramping of the hands and minimize shoulder/arm muscle strain
- In 1987 Grandjean suggested 25° angle between key halves, a distance of 950 mm between G and H, and a lateral slope of 10°



BAT Chord Keyboard

- Uses a “chord system” for input
  - Every letter or character is generated by pressing a certain set of keys
- Difficult to learn because there are no “obvious” inputs
- Requires very little arm / hand movement to use

http://www.enablemart.com/

Half-QWERTY Keyboard Variant

- Can be used as a standard QWERTY keyboard, or in one-handed mode that uses only the left side
- Much easier to learn because of standardized layout and ability to use it normally while learning
- No size advantage

http://www.half-qwerty.com/

Projection and Sensor Keyboards

- Futuristic design
- Extremely Portable
- Allows "air-typing" – the ultimate in touch typing
- Senses finger position and key press
- Interface with both PDA's and standard computer

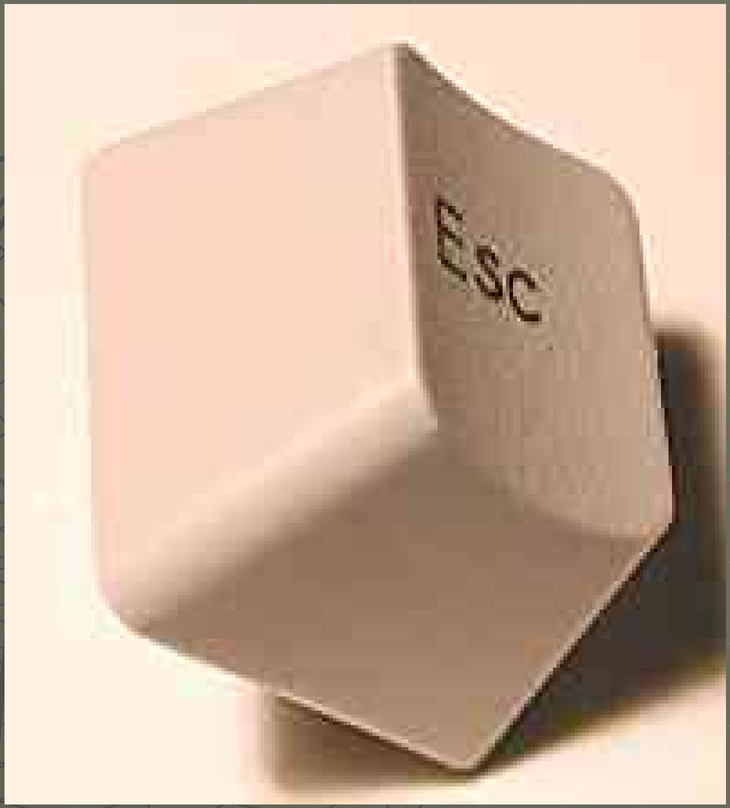
Size and Shape of the Keyboard

- Standard Dimensions:
  - Width: 6 ½ inches
  - Length: 18 inches
  - Front height: 1 inches
  - Back Height: 1 ½ inches
- Can differ with style, layout, type, and make of keyboard

Keyboard Components

Keycaps

- The "keycap" or "Key Top"
  - The part of the keyboard that you strike with your fingers while typing
  - Removable / replaceable on modern keyboards

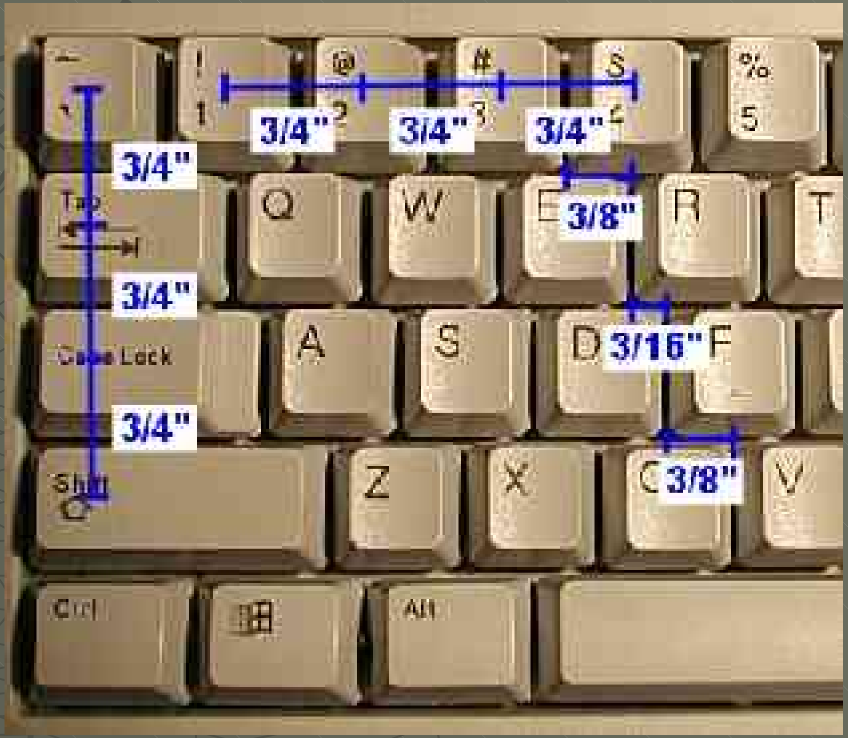


Keycap Size

- Keycap size is fairly standard
- The top area that you strike is approximately .5 square inches
- The cap tapers down to approximately .75 square inches at the base

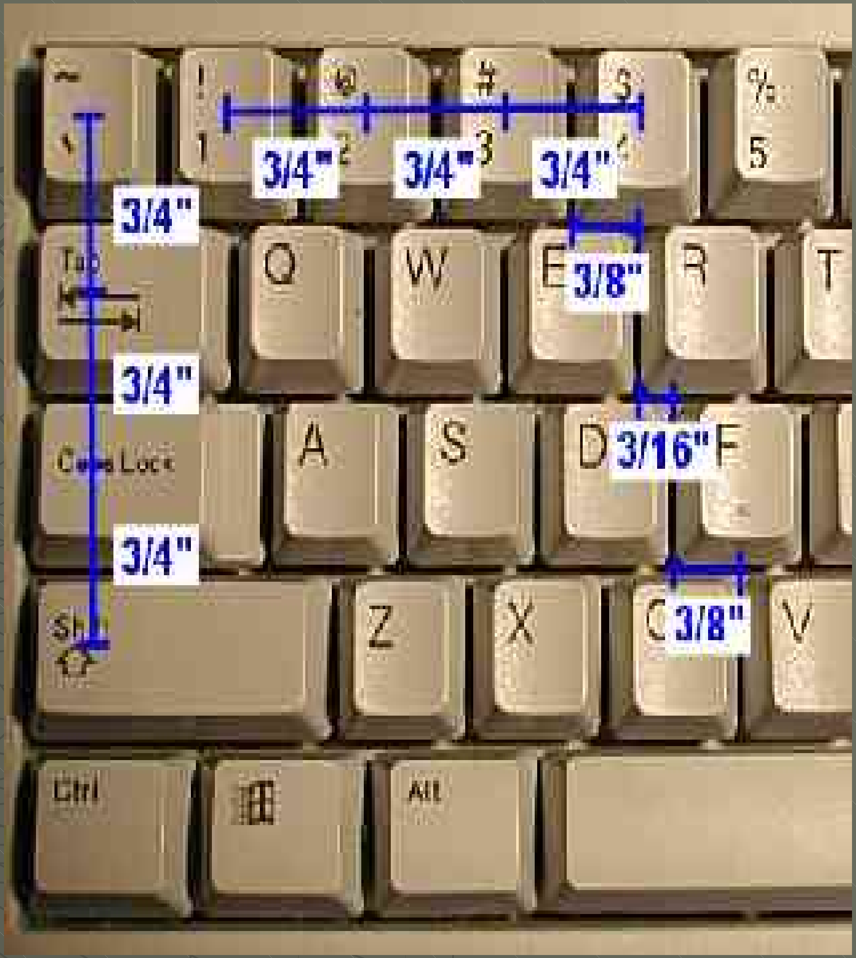
Keycap Spacing

- On a horizontal line, keycaps are approximately .75 inches center to center
- Each row is approximately .75 inches center to center



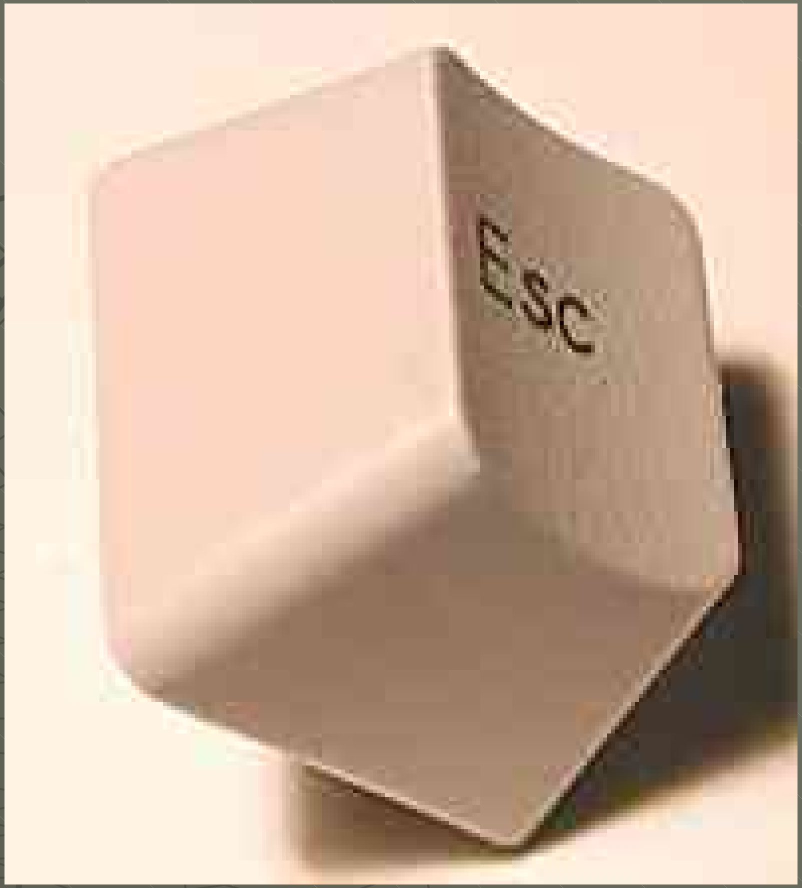
Keycap spacing (cont'd…)

- The keyboard rows are staggered
  - Numbers to qwerty row – 3/4" shift
  - Qwert to asdfg row – 3/8" shift
  - Asdfg to zxcvb row – 3/4" shift
- Same for most keyboards so users don't have to relearn every time they type



Keycap Shape

- Shaped like a square pyramid with the top cut off (3D rhombus?)
- Top surface is bent and rough / textured



Keycap Travel

- The distance a key moves when struck is called the "travel".
- Varies from keyboard to keyboard
- Our keyboard travel is approximately 1/10th of an inch
- Long travel is considered more desirable by typists
- Short travel is usually used in applications where space is limited
  - Laptop keyboards, etc.

Key Switches

- The mechanical component below the keycap
- Initiates electrical contact with each keypress to send a signal to the keyboard processor
- Key switch standards vary by keyboard and manufacturer
- Most common kind is the membrane-based key switch, consisting of a flexible rubber membrane and two sheets of electrical contact points.
- When pressed by the keycap, the membrane closes the two electrical contacts to complete the circuit specific to that keycap and its associated key.

Key Switch Variation

- Membrane (our keyboard)

- Rubber Dome

- Other types:
  - Foam and foil
  - Capacitive

- Mechanical

Key Switch Comparison

- Mechanical: durable but expensive; high tactile response
  - When key is pressed, key cap depresses spring and closes metal contacts
- Foam-&-Foil: inexpensive but not practical for heavy use; soft tactile response
  - Foil bubble is deformed downward with the action of the keypress to close a metal contact
- Membrane: low tactile response; resistant to spills and debris
  - Rubber membrane providing tactile resistance
  - Consists of three layers of plastic: upper and lower levels have embedded circuit traces. Middle layer has holes under each key
  - With keypress, contact on upper level passes through hole in middle level to meet contact in lower level
  - Most common key switch type
- Rubber dome (carbon contact): decent tactile response.
  - Much like membrane type, but without plastic layers
  - Carbon on underside of dome closes contact
- Capacitive: no tactile response
  - The capacitive switch runs on the concept that two pieces of metal in very close proximity of each other are able to hold an electrical charge
  - The processor in these keyboards determines which key is pressed by comparing the capacitances of the connected lines

Key Pitch

- Key pitch is the angle of the top of the key cap relative to the surface upon which the keyboard rests
- Pitch increases from the bottom row of the keyboard to the top row, as illustrated above
- Standard full sized keyboards have substantial pitch, while laptops typically lack any pitch whatsoever due to their compact size and flat keyboards
- Pitch aids in typing comfort by enabling the typist to place hands in a position that lessens pressure on the wrists

Keyboard LED's

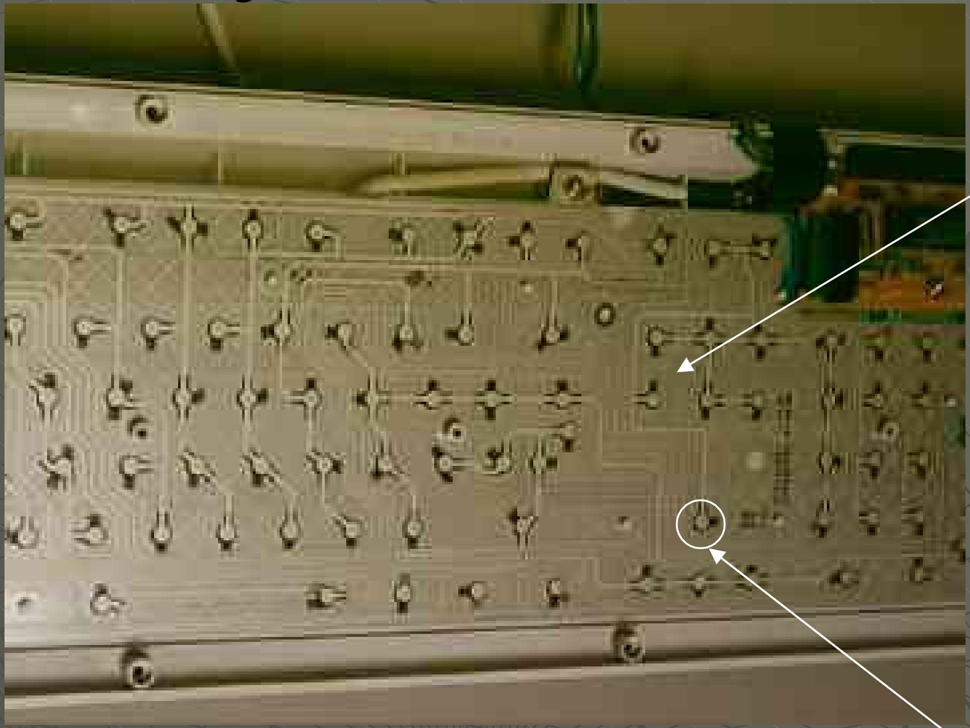
- An LED (Light Emitting Diode) uses electric current through a semi-conductor to generate visible light
- Used to indicate the status of the toggle keys
  - Num Lock
  - Caps Lock
  - Scroll Lock
- Can be toggled by the computer or by pressing the keys on the keyboard

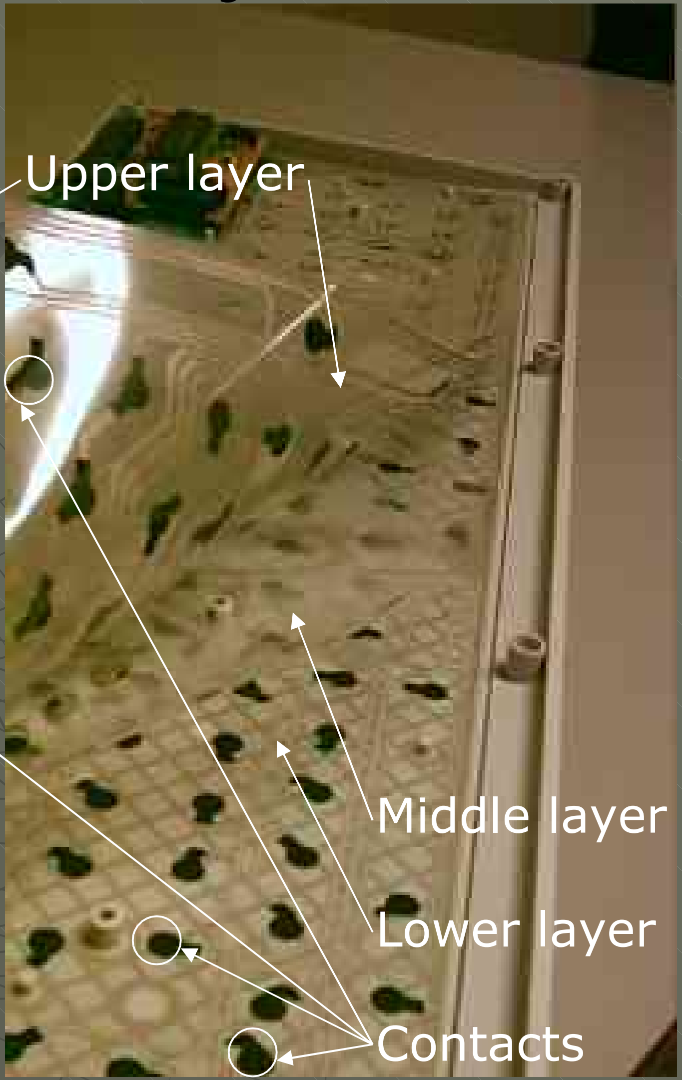
Research

Electrical Aspects

- Key Contacts
- Key Matrix
- Keycode Processor
- Keyboard Output as Signals: The Clock and Keycodes
    - Interpretation of Keyboard Output Signals
- Make / Break Codes
- Port pinouts

Key Contacts and Key Matrix

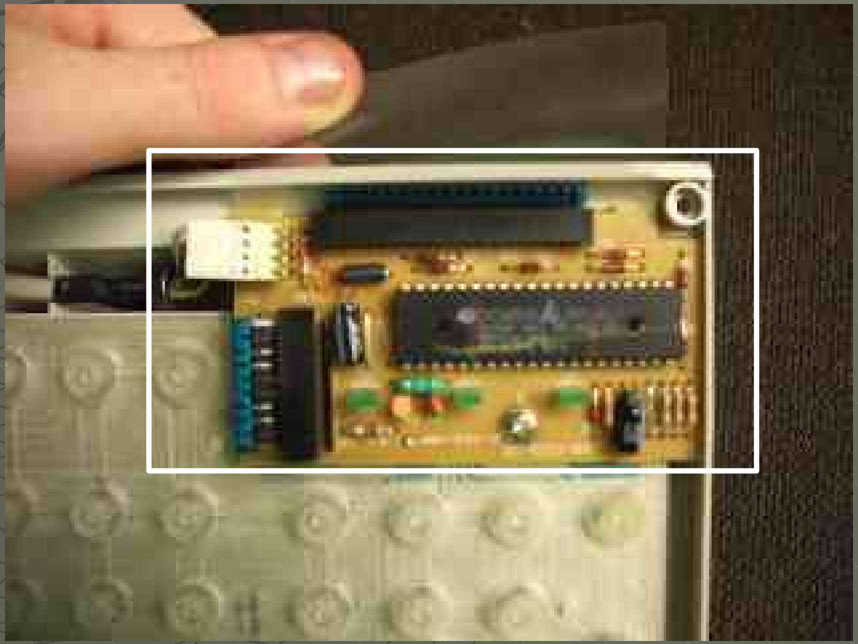




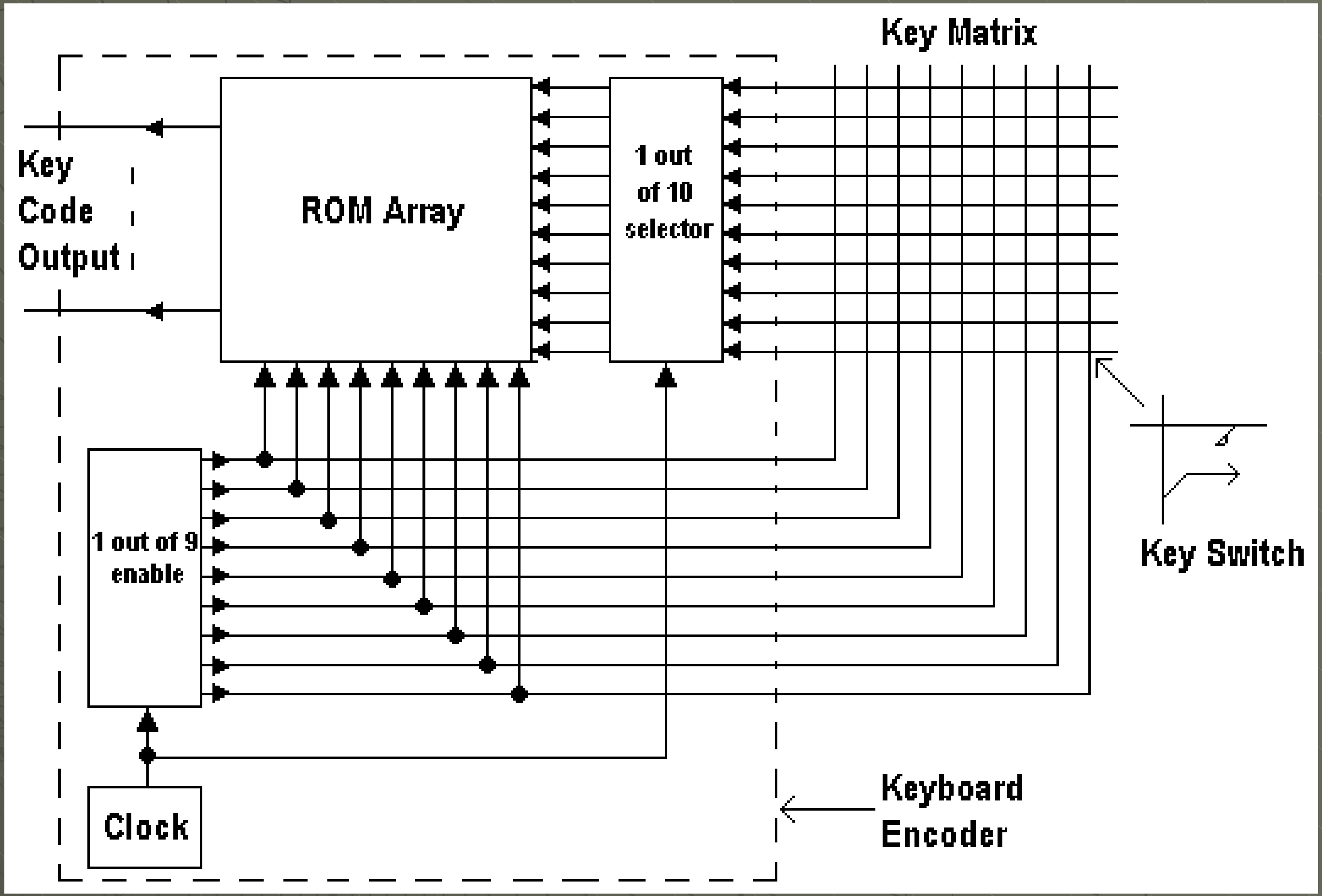
Key contacts as pictured are part of a matrix of circuit traces (those lines that connect the contact points). This key matrix is connected to the Keycode processor; it determines where and which key was pressed in the matrix, as well as the Keycode to send to the computer.

The Keycode Processor

- Determines the key pressed in the key matrix
- Generates a pair of signals that can be interpreted by the computer
- These signals are called the Keycode and the Clock
- The computer also sends data to the keyboard. (software can set num lock, etc). Therefore, communication is bi-directional.



Key Processor Schematic



The Keycode

- Each individual key has its own unique code associated with it
  - For example: the Keycode for the 1 key on the keypad is different from the 1 key above the letters

The Clock

- When a key is pressed (and the Keycode is subsequently generated by the Keycode processor), a second part of the circuit is activated: the timer, or clock.
- The clock is synchronized with the Keycode sent to the computer and provides a frame of reference for interpreting it.

The Clock as a Signal

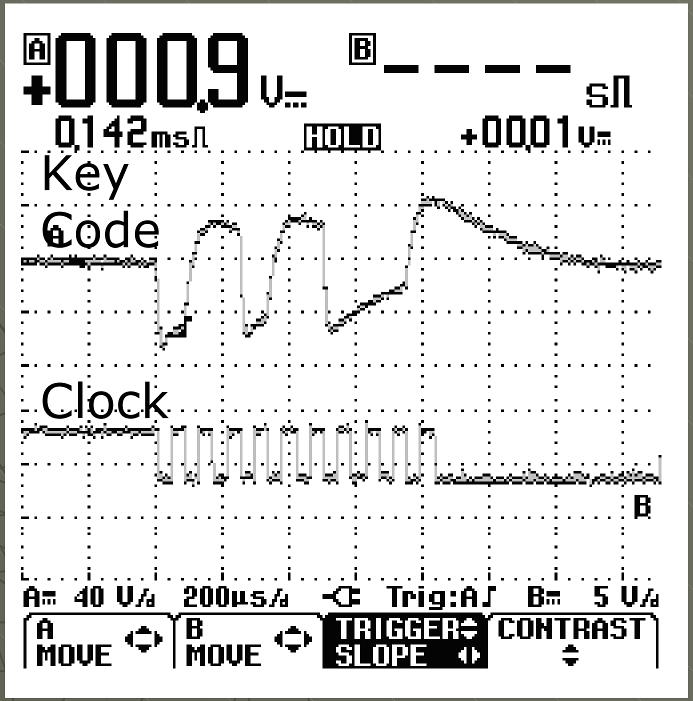
- The clock output is a digital signal that is the same no matter which key is pressed
- It consists of “ticks” which coincide with the Keycode sent to the computer from the Keycode processor

The Keycode as a Signal

- The Keycode is an analog signal that can be broken down into a stream of bits by the computer by comparing it to the “ticks” of the clock signal

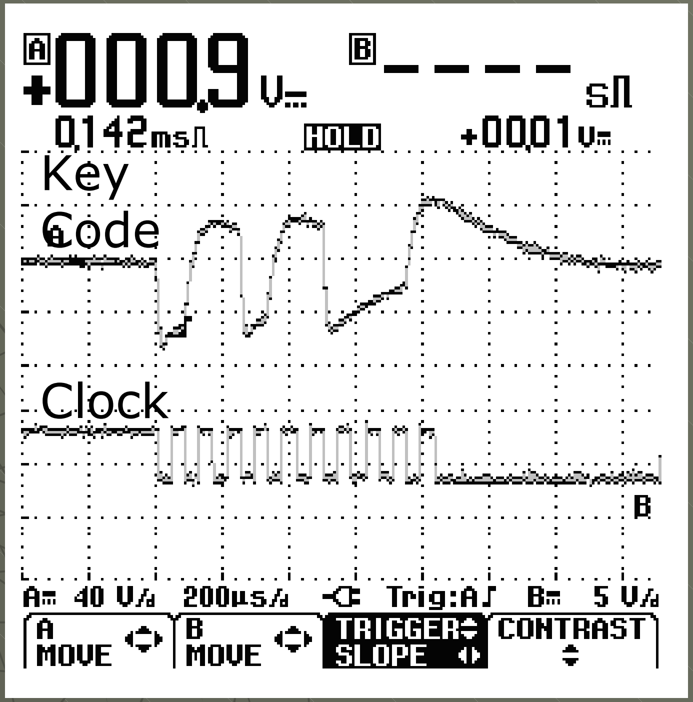
Keyboard Output as a Combination of Signals

- The keyboard signals are transmitted to the computer as pulses of electricity to be converted into a binary number
  - The Keycode signal ranges from +4.8vDC to +5.5vDC
  - The Clock signal ranges from +2vDC to +5.5vDC
- One tick of the clock consists of a drop in signal and a jump back to its original value; 11 evenly spaced ticks occur per keypress
- The Keycode signal is less regular and its up or down state is determined only by its voltage in comparison to +5vDC
- The keycode is divided into 11 sections, each corresponding to one tick of the clock



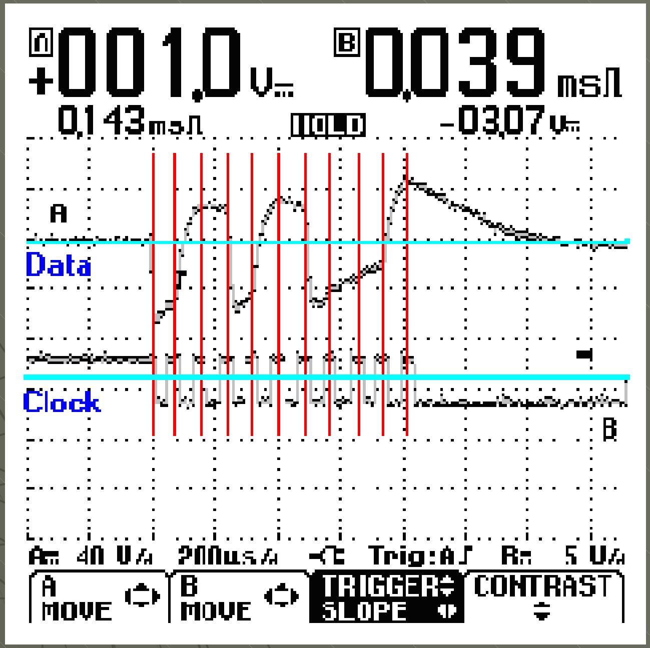
Keyboard Output as a Combination of Signals, cont'd

- For every tick of the clock, the computer reads the voltage of the keycode
  - If the voltage is above +5vDC, the computer reads a binary 1
  - If the voltage is below +5vDC, the computer reads a 0



Keyboard Output as a Combination of Signals, cont'd

- In the example shown at right, the red lines are aligned with each of the 11 ticks of the clock
- The blue line on the data signal signifies +5vDC
- Looking at each of the 11 subsequent divisions of the data signal, the following binary number is translated: 01101100011



Interpreting the Binary Translation of the Keycode

- In the previous example, the binary number 01101100011 represents a packet of information sent to the computer for that keypress
- This packet consists of 11 binary bits, including a leading "start bit" which is always 0, a "stop bit" which is always 1, and a "parity bit" which corresponds to the number of 1's in the binary data, and is used for error checking
  - The parity bit is the bit just prior to the stop bit
  - A parity bit of 1 implies that there are an even number of 1's in the binary data, while a 0 implies an odd number of 1's
- The remaining 8 bits represent the character of the key pressed and are entered backwards

Interpreting the Binary Translation of the Keycode, cont’d

- Further continuing the previous example, the binary number 01101100011 can be broken down into the following four parts:
  - Leading 0 (start bit)
  - Ending 1 (stop bit)
  - Parity bit 1 implying an even number of 1’s in the data
- Subtracting the start, stop, and parity bits from the binary data, the remaining number is 11011000
  - These 8 bits is a backwards representation of the keycode
  - The actual binary number for the keycode is 00011011
- Keycodes are usually read in hexadecimal notation
- Converting this binary keycode 00011011 into hex, we obtain the number 1B
- This hexadecimal keycode can now be converted via “scan codes” by the computer into an actual useable character, displayed as pure text

Scan Codes

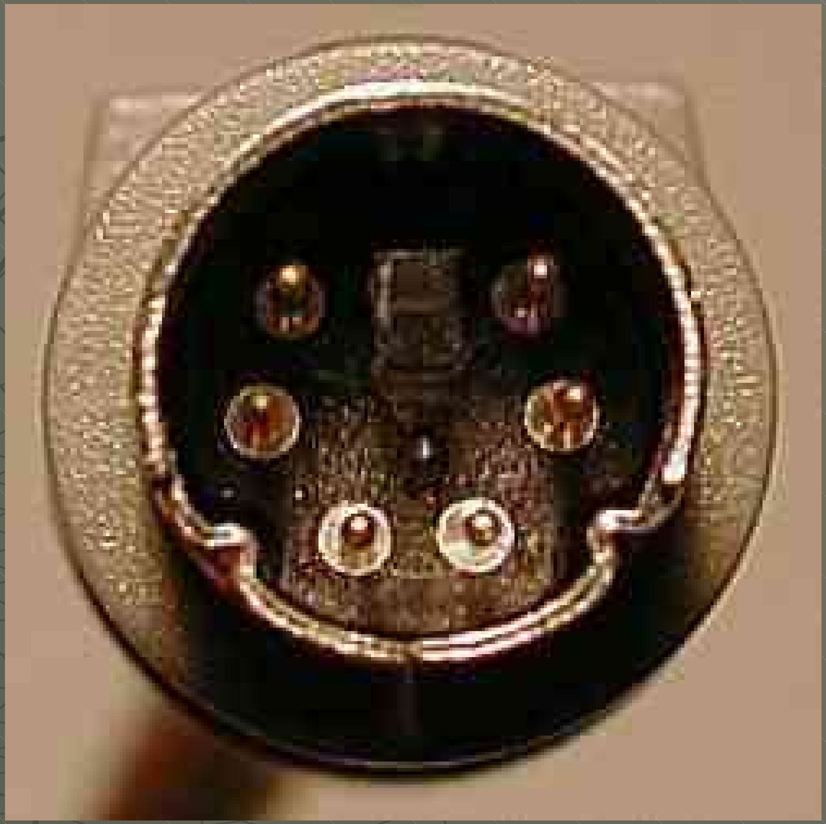
- Binary keycodes converted into hex are considered "scan codes"
- For each keyboard, there are several scan code charts that can be used to translate the scan codes into characters
  - Scan code charts are specific to language, geographical location, and keyboard layout
  - They are interchangeable
  - They may be accessed and used by an Operating System as required
- There is a subset of scan codes considered to contain "make codes" and "break codes"
  - Make codes are simply the scan code hex value and imply that a key has been pressed down
  - Break codes are generally the scan code hex value plus 80 (in hex, not decimal
  - Make and Break codes are especially used for handling multiple-keypresses to determine which keys have been pressed and which have been released
    - An example: pressing Ctrl+Alt+Del would result in the transmission of the make codes for all three, but only the break code for Del
    - The computer then interprets the break code for Del, and the lack of break codes for Ctrl and Alt to mean that all three were pressed in combination

Scan Code Examples

Key Number	Scan Code	Key
1	0E	`
2	16	1
3	1E	2
4	26	3
5	25	4
6	2E	5
7	6	6
8	3D	7
9	3E	8
10	46	9
11	45	0
12	4E	-
13	55	=

Port Pinout

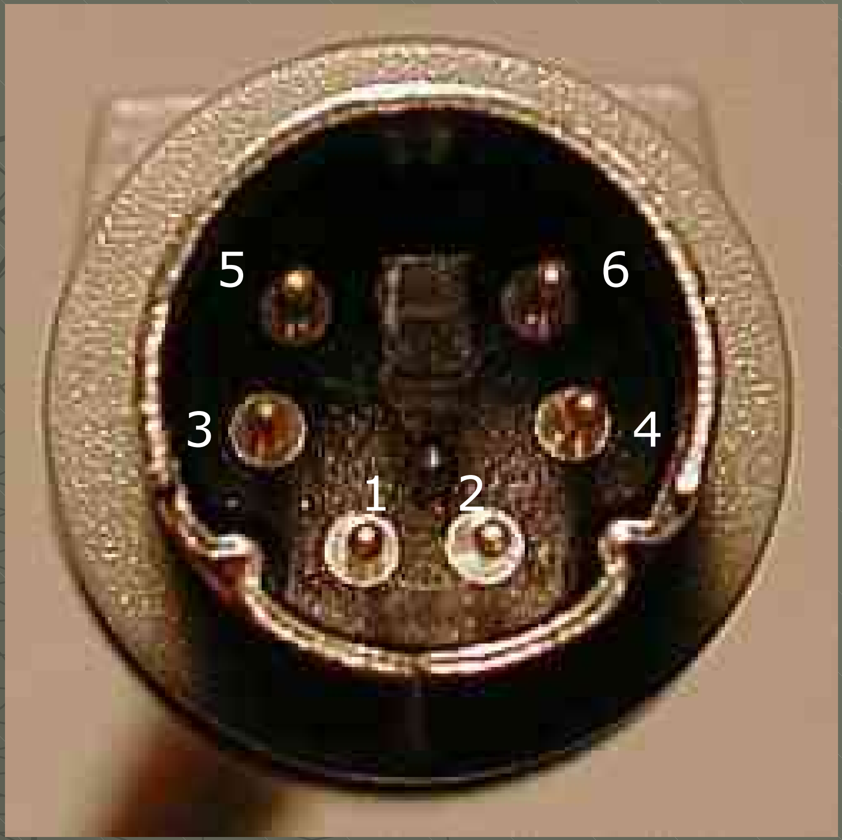
- Most modern keyboards use a PS/2 or "mini-DIN" connection
- Each pin in this port is associated with a number, 1-6, from left to right



Port Layout (continued…)

PIN Signals:

- 1 – Keyboard Data
- 2 - not connected
- 3 – Ground
- 4 – Power (+5V)
- 5 - Keyboard Clock
- 6 – not connected



The Fun Begins: Disassembly

Disassembly Procedure

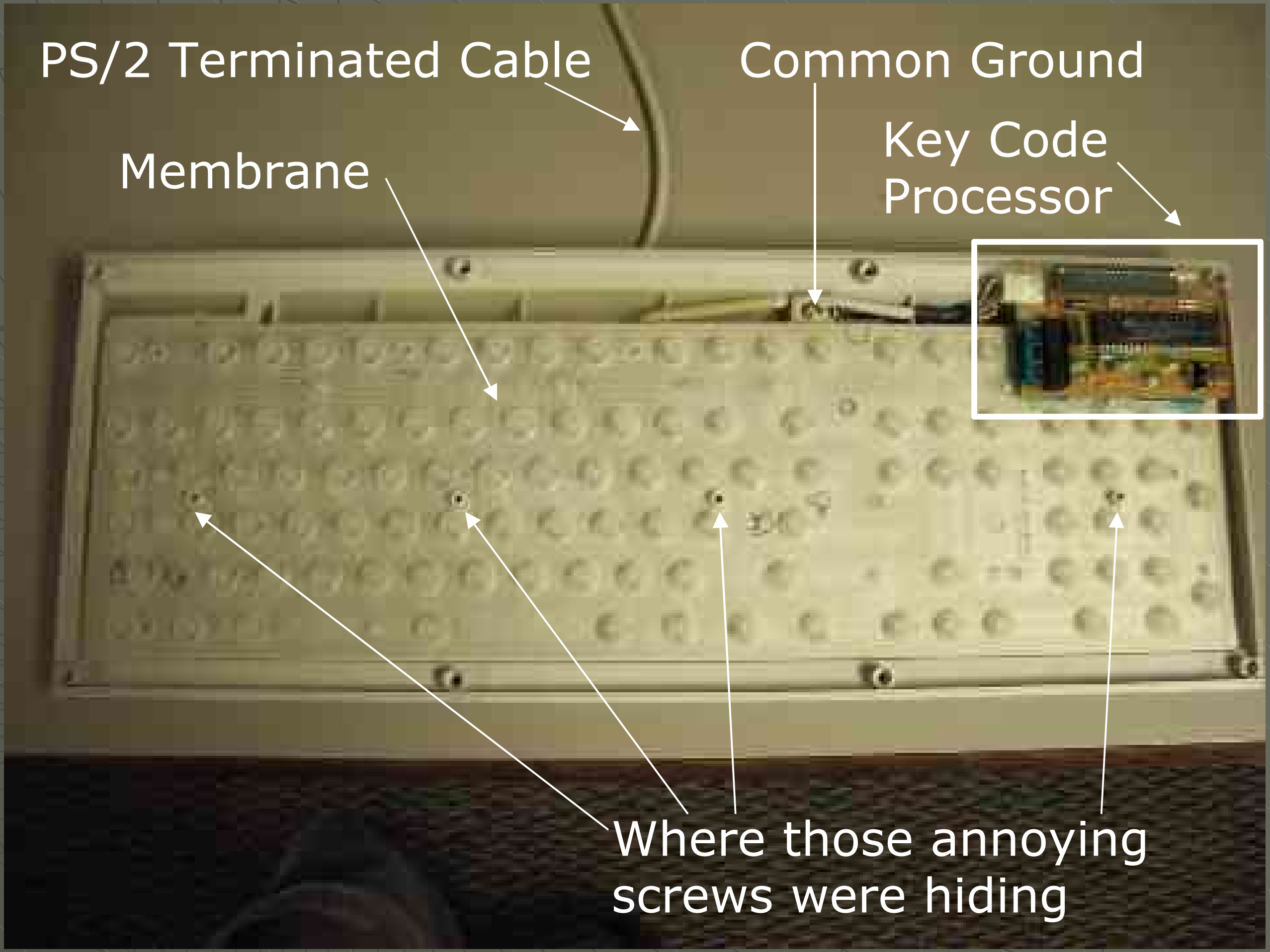
- Remove keycaps with simple lever
- Remove screws below keyboard and under keys
- Remove top half of keyboard
- Remove membrane

The Keyboard Itself

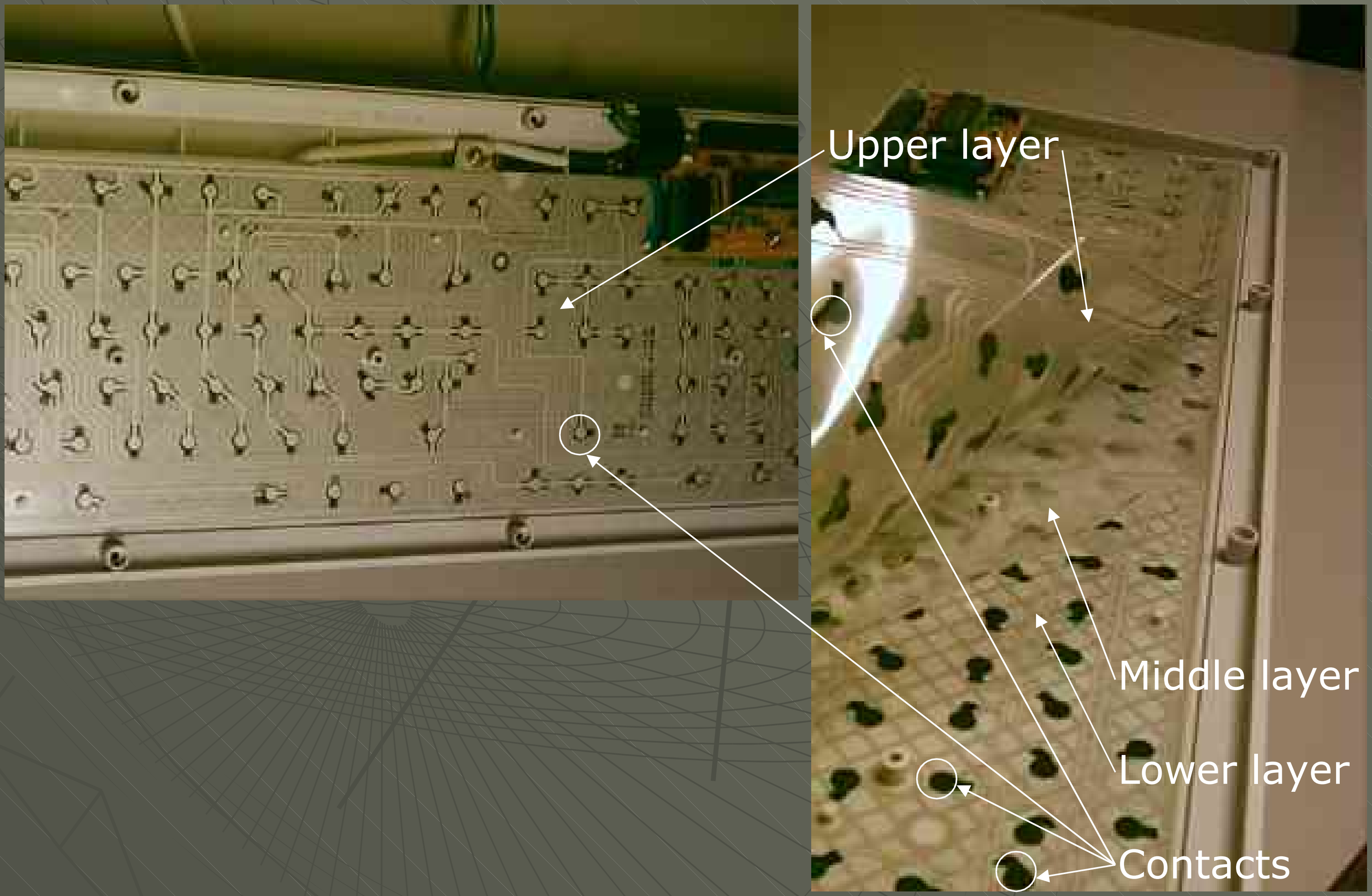
Keycap Fun

Wouldn't you?

Under the Hood



Revisiting The Membrane: What Lies Beneath

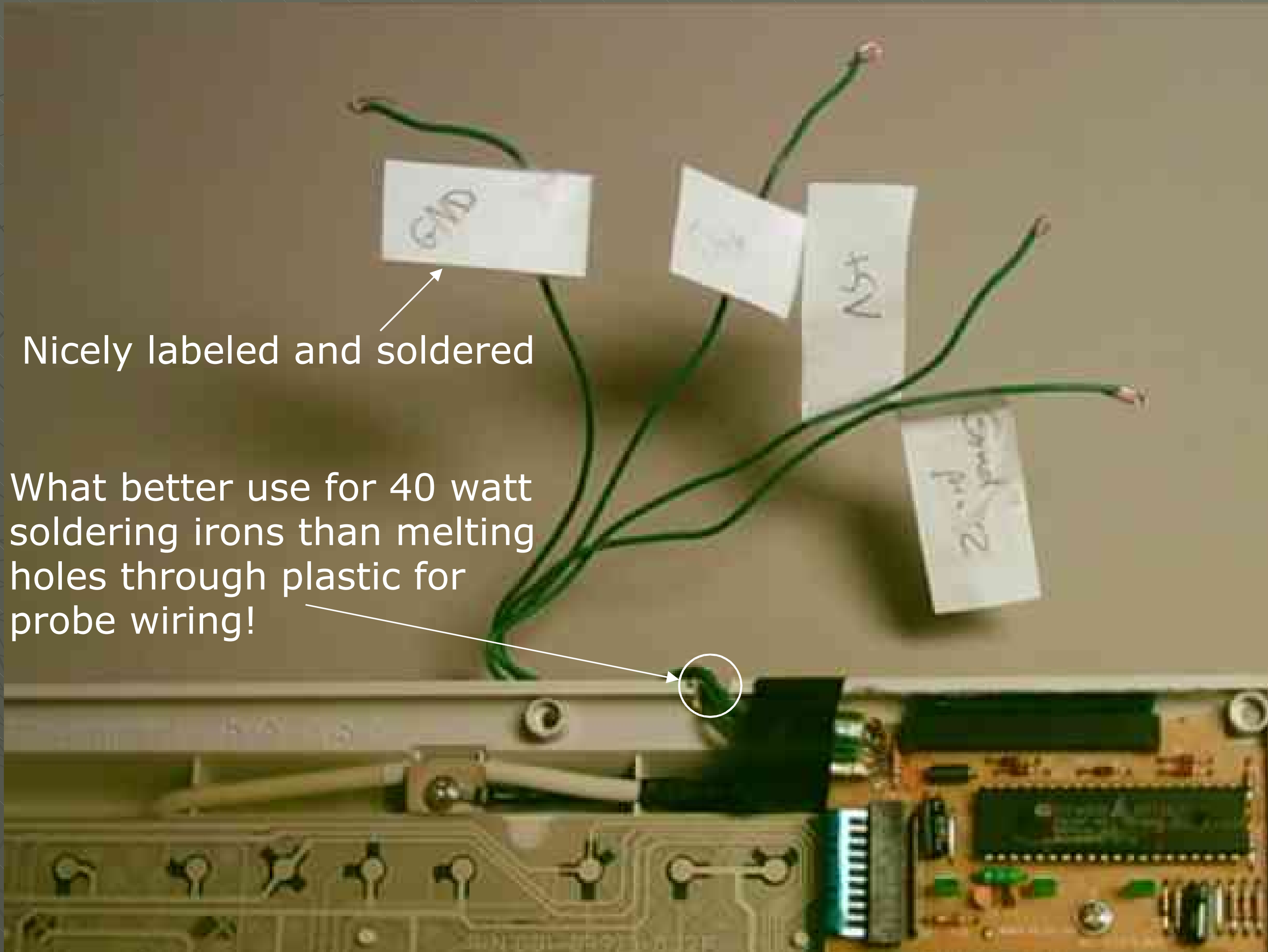


Like a Visit to the Proctologist, The Probing Begins

Probing Procedure

- Using a lovely Fluke Multimeter and its lovelier continuity feature, trace PS/2 port pins to the connector on the key code processor PCB
- Solder probe extension wires to connector pins
  - Be careful not to short out the wires or melt the board that the pins are connected too
  - Be ESPECIALLY CAREFUL not to damage the chip
- Referring to PS/2 pinout, label each wire with an appropriate tag
- Replace keyboard cover, screw back in, reattach keycaps, and get to work with an oscilloscope


That was painful. You'd think dorm lounges would be better lit.


Nicely labeled and soldered
What better use for 40 watt soldering irons than melting holes through plastic for probe wiring!

Signal Probing And Analysis with the Oscilloscope

The Oscilloscope

- We used Fluke 123 “Industrial Scopemeter” oscilloscope for signal scanning in this project

PS/2 pins 3 and 4 (Ground & Power)

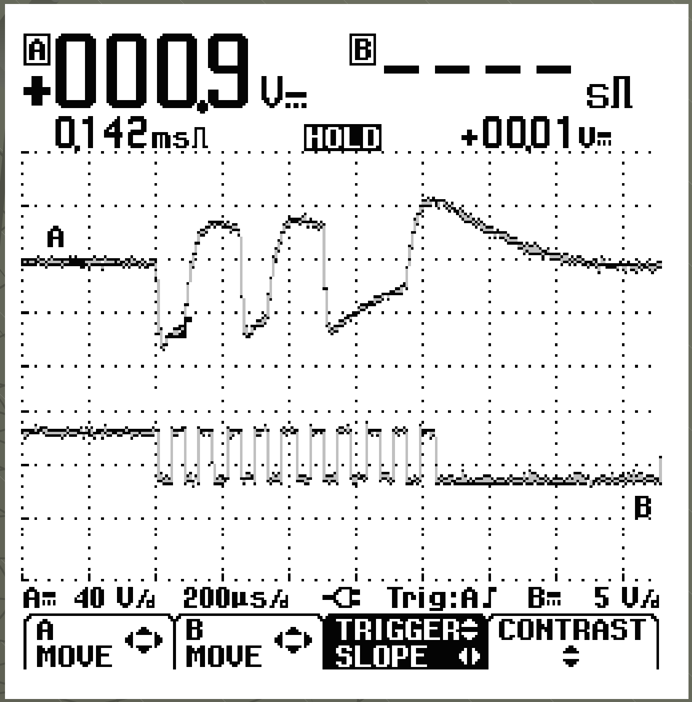
- The ground line is simply a ground connection for the keyboard and is the basis for all signal technology
  - Think of it as a base reference to which all other signal data can be compared to
- Power line (+5vDC) is the wire that the computer uses to provide the keyboard with electricity to operate
  - It is +5vDC because it carries a signal 5 volts above the true ground line, 0vDC

The Ground

- Connect the “common ground” (center) wire of the oscilloscope to the +5vDC wire on the keyboard
  - We used the +5vDC line as the basis for all our measurements (the “ground”) because it was closest to the reference voltages on the keyboard data line

The Keyboard Data

- Connect the signal A input on the oscilloscope to the "Keyboard data" line on the keyboard:
  - +4.8 volts to +5 volts = a 0 bit
  - +5 volts to + 5.5 volts = a 1 bit
- Adjust the oscilloscope voltage trigger so that the "A" signal is shown near the top of the screen when the key is pressed
- Connect the Fluke 123 Scopemeter to the test-bed computer using the special serial-port adapter. Then, fire up the FlukeView software package and take this picture
- Pretty isn't it?



The Importance of a Steve

- In order to properly read the signal from the clock on the keyboard, it is necessary to place a strongly resisting Steve between the oscilloscope and the keyboard clock wire.
- The electrical resistance of the needed Steve (or wire as the case may be when one lacks a Steve) is about 1 megohm.
- (The Fluke was throwing back some residual voltage *back* into the system and messing with the signal..wish it hadn't taken so long to figure that out)

The Clock

- Connect signal B input to a 1 megohm resistor (or resistive wire) and the other end of the resistor to the "Clock" line
  - You need the resistor in there to dull down the signal being carried through the wires, or else when you connect the oscilloscope to the circuit, the feedback between "clock" and "+5" (the ground) will mess up the keyboard's data signal

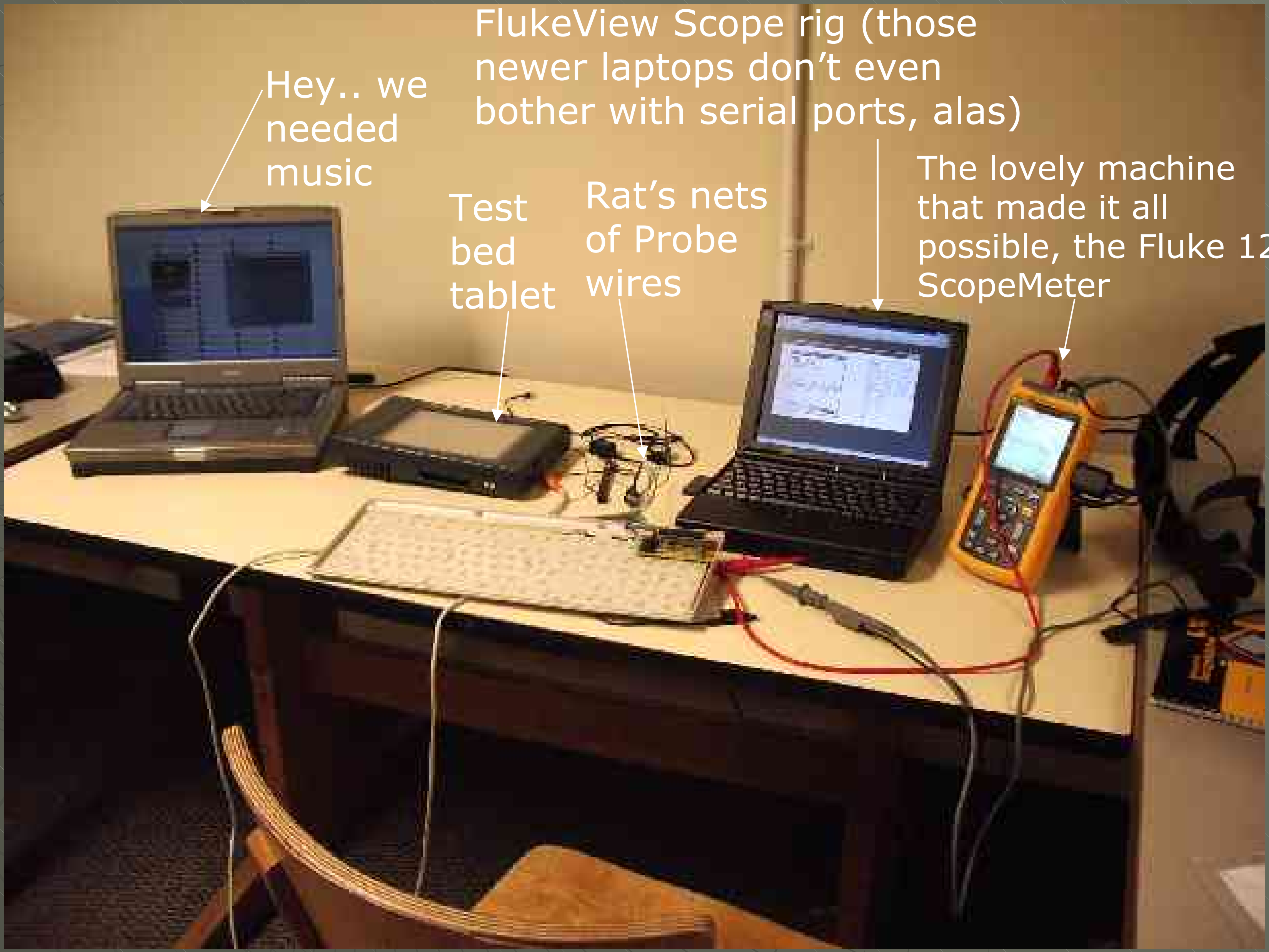
Oscilloscope Settings

- Input from both probes is on and set to read pulses
- The voltage amplitude is set to 40 volts DC on input A and 5 volts DC on input B
- The time width on the graph is set to 200 μ-seconds

Getting the scan codes

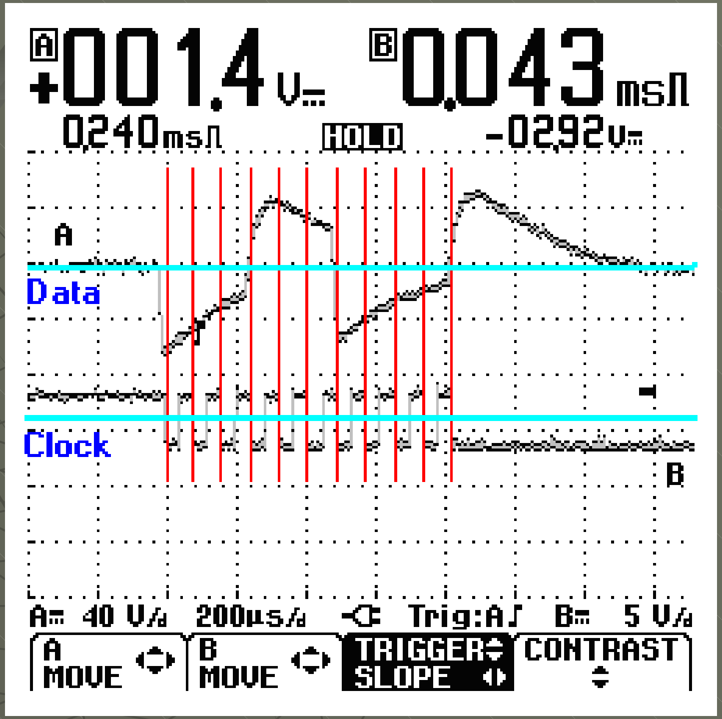
- Press the key on the test keyboard of which one desires the scan code
- Hold it down until the signal stabilizes on the oscilloscope screen
- Press the “Hold” button on the scope
- Take a screen shot of the scope using the FlukeView software
- Save the screenshot
- Overlay screenshot with grid to help interpret the signals

We Invaded!


FlukeView Scope rig (those newer laptops don't even bother with serial ports, alas)
Hey.. we needed music
Test bed tablet
Rat's nets of Probe wires
The lovely machine that made it all possible, the Fluke 123 ScopeMeter

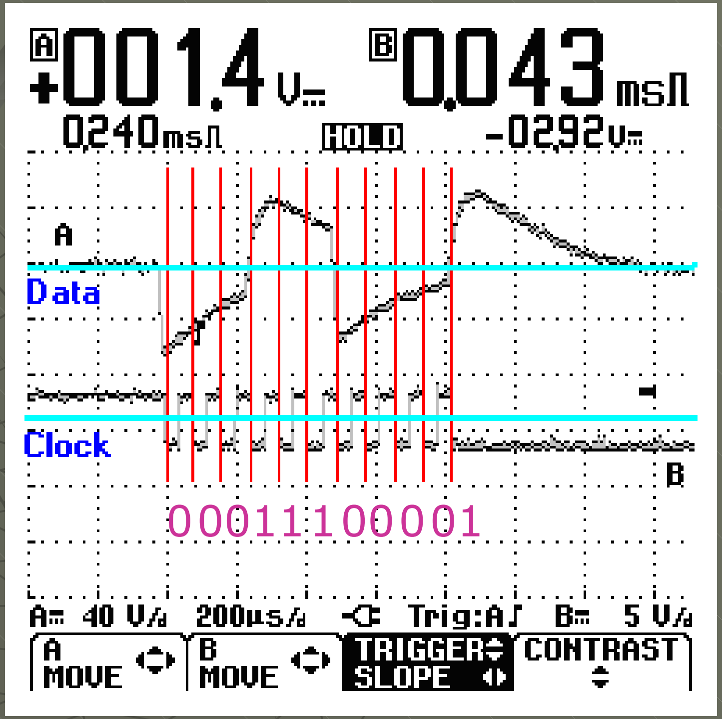
Results

- Following the above procedure (the probe setup, the use of Steve, the screen capture and overlay of the signal, the image to right of the letter A was generated.

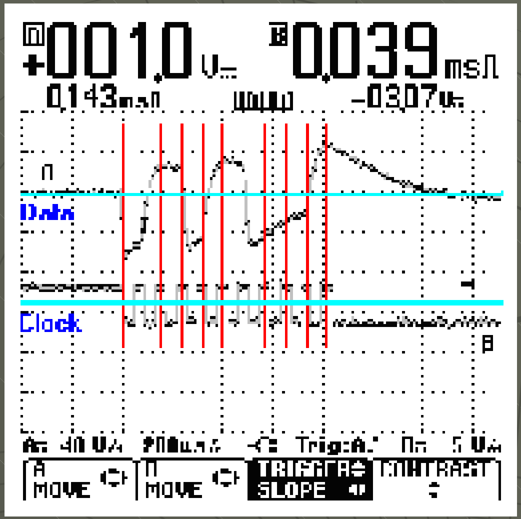


Results: Meaningful Information from Oscilloscope Readings and Binary

- The image shown to right depicts the signal for the letter A.
- Data signals below the +5vDC (top blue) line are binary 0's; above, 1's.
- Using the clock as reference, one can determine the scan code of A to be: 0|00111000|0|1 (where |'s separate start, stop, and parity bits from the scan code itself)
- 0 – Start Bit
- 1 – Stop Bit
- 0 – Parity Bit
- What remains: 8 bits of data – 00111000
- Following PS/2 specifications, reversing the 8 data bits – 00111000
- Converted to hex: 1C
- Referring to the scan code, chart 2 for our 101/102 key keyboard, this does indeed produce the correct character interpretation
- Thus: A=00111000=1C



Additional Results: Signal Analysis of the Home Row and Other Oft Used Keys

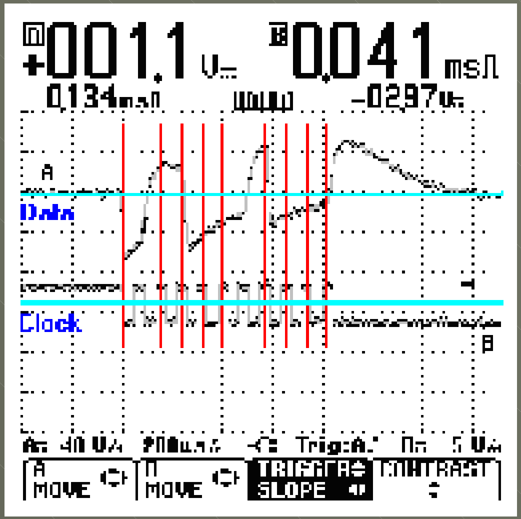


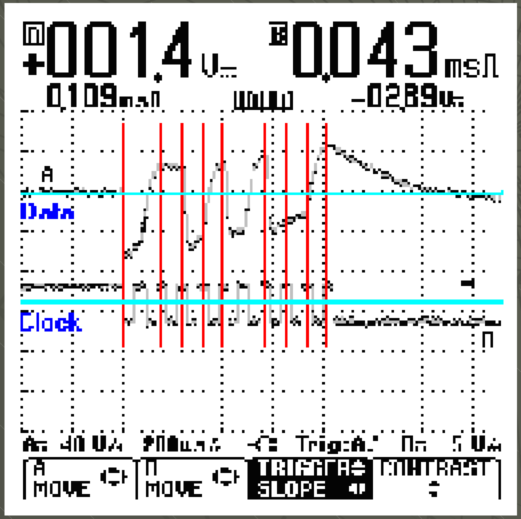
←S=0|11011000|1|1

=000110=1B=S

D=0|11000100|0|1

=00100011=23=D→



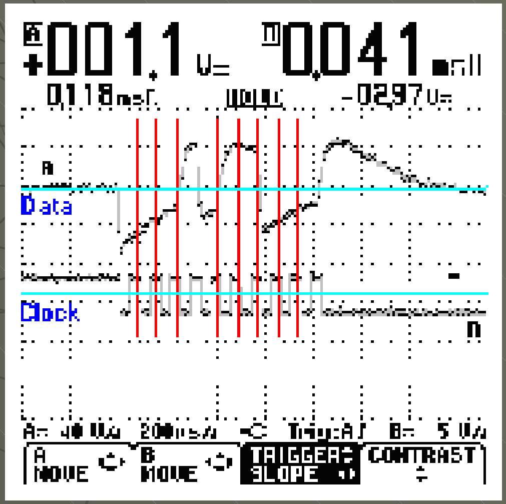


←F=0|11010100|1|1

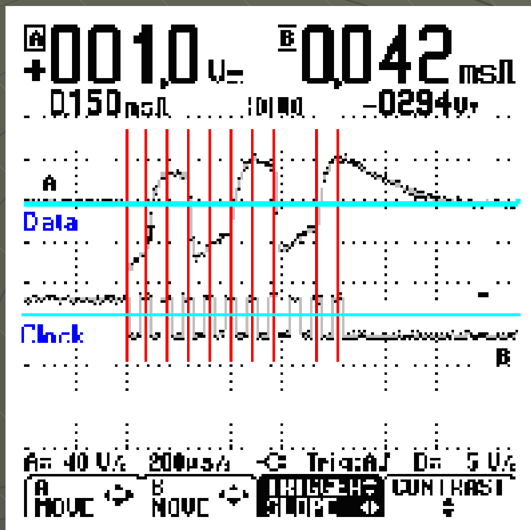
=00101011=2B=F

G=0|00101100|0|1

=00110100=34=G→



Additional Results, cont'd

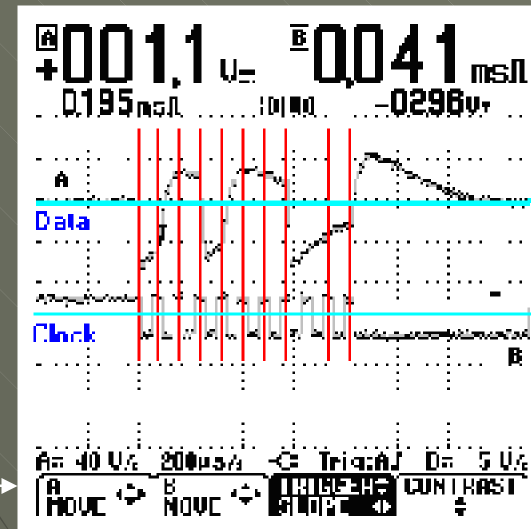


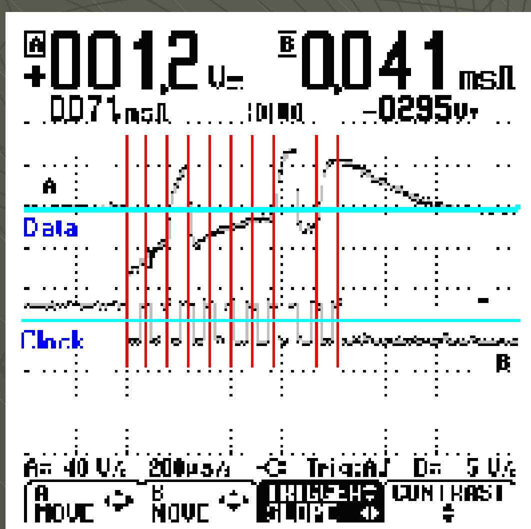
H=0|11001100|1|1

=00110011=33=H

J=0|11011100|0|1

=11011100=3B=J



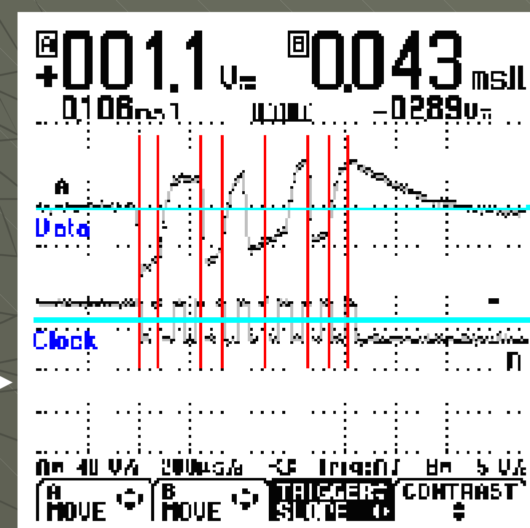


K=0|01000010|1|1

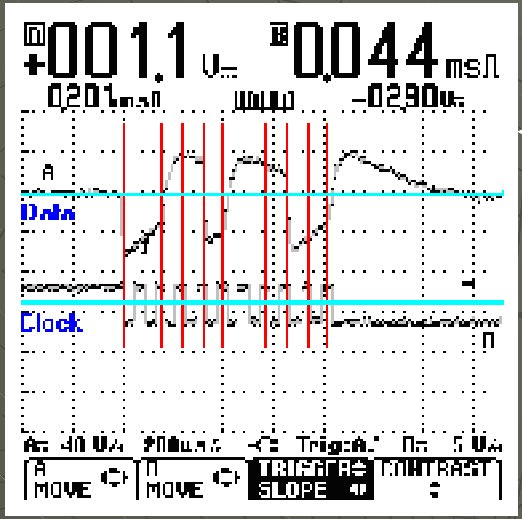
=01000010=42=K

L=0|11010010|1|1

=01001011=4B=L



Additional Results, cont'd

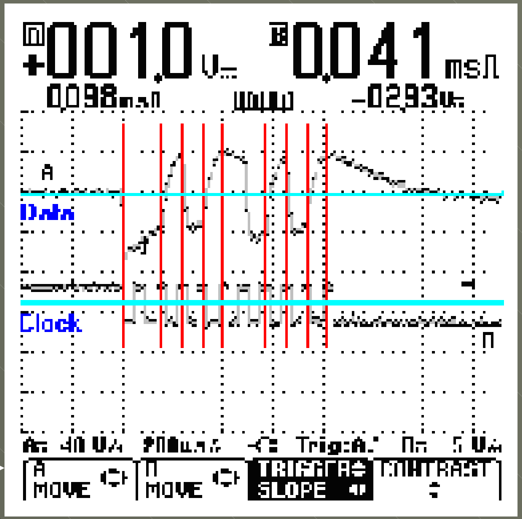


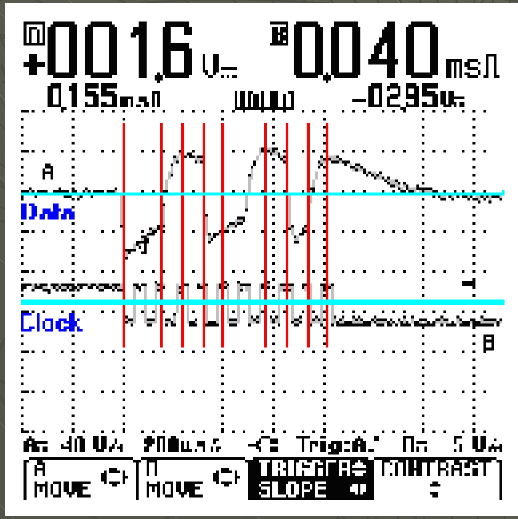
Esc=0|01101110|0|1

=01110110=76=Esc

Entr=0|01011010|1|1

=01011010=5A=Entr



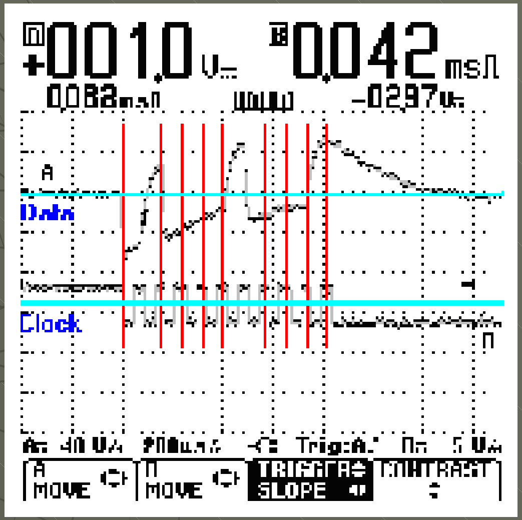


Bkspc=0|01100110|1|1

=01100110=66=Bkspc

Left Alt=0|10001000|1|1

=00010001=11=Left Alt



Summary

- In the probing of other signals, our results directly concurred with scan codes published for our variety of keyboard
- From the source we used at http://cma.zdnet.com/book/upgraderepair/ch09/ch09.htm; we used Set II

Surprises!

- There were screws located under keys
- Very dirty under keys
- Projection keyboards are cool (not really a surprise but still cool…)
- Mix up with PS/2 cable wiring pattern (numbering sequence isn't intuitive)
- When we connected the keyboard and oscilloscope to the computer in one of our first arrangements, we could press keys on the keyboard and they wouldn't show up on the screen. Then when we disconnected the oscilloscope, all of the characters we pressed were sent to the computer
  - This implies that the chip on the keyboard contains a buffer to hold characters as they are being sent.
  - After testing we discovered that the buffer holds up to eight characters at once.

In Conclusion

In the making of this report the entire process of converting key presses to signals, binary data, and back to text occurred 10’s of thousands of times in a process that is wonderfully automated and we never have to handle ourselves, but now understand anyway… Aren’t computer’s wonderful?

Sources

- www.pcguide.com
- Kroemer, K.H.Eberhard “Human Engineering the Keyboard” Human Factors vol. 14 1972
- Baber, Christopher Beyond the Desktop Academic Press. San Diego, California 1997 pp. (25-46)
- http://www.maltron.com
- http://www.frogpad.com/
- http://www.enablemart.com/
- http://www.half-qwerty.com/
- http://www.alpern.org/weblog/stories/2003/01/09/projectionKeyboards.html
- http://www.soc.staffs.ac.uk/wf1/FCS/5peripheralsLectureNotes.ppt
- IBM PS/2 A Reference Guide, TJ Byers, Intertext Publications, McGraw-Hill Book Company, 1221 Avenue of the Americas, NY, NY 10020, 1989 ISBN: 0-07-009527-2