How I Decoded a $5 Surplus Card Reader Before the Internet

- Hardware
- Encoding Standards
- Demo

Hardware - Reader

- Purchased from Active Surplus in Toronto over 25 years ago.
- Label: MSR-211 ND 497000-0171 NIPONDESNO CO LTD. JAPAN
- Five wires, all different colours

Hardware - Power

- First, identify wires for power and ground.
- Typically, look for polarity of capacitors or IC power and ground pins.
- Got lucky: PCB is labelled "5 V" and "GND" !
- Traced pcb GND to BLACK wire and "5 V" to RED wire.

Hardware - Signals

- Next, traced remaining wires to transistors which are connected to ground. i.e. They are open-collector outputs.
- Pull-up resistors will be needed for digital inputs to microprocessor. Remeber, this was the 1980's and there were no Arduinos!
- The system I was interfacing to was a Z80 Multiflex, S100 bus system running with a 1 MHz clock. Yes, that was state-of-the-art for the hobbyist.

Hardware – Card Detect

- Light sensor across slot so one of the wires must be a CARD DETECTED output.
- Connected a voltmeter to each of the three wires in turn and used a card to interrupt the sensor. Determined that it was the GREEN wire.

Hardware – Data Signals

- In those days, BYTE magazine and books from Radio Shack were your main source.
- From those, the remaing signals were most likely DATA and STROBE, but which is which?
- A voltmeter does not help here. Anologue meters could not read pulsed signals as they averaged out as a shaky needle and narrow pulsed signals would not move the needle at all.

Hardware - Oscilloscope

- To debug the Z80 system (built from a kit), what was the slowest oscilloscope needed for a 1 MHz clock?
- Nyquist frequency is double what you want to measure so a minimum of 2 MHz. Fortunately two factors helped choose a higher frequency:

1) My dad was an Electrical Engineer and told me that the practical frequency should be 10 times the highest frequency you want to measure, and
2) The lowest frequency oscilloscope that was available was the Hitachi V-152F at 15 MHz for $350

Hardware - Waveforms

- One signal did not seem to react to card swipes.
- The other signal gave many pulses of varying patterns so THIS signal is most likely the DATA line. It is the ORANGE wire.
- If the STROBE signal is defective it could be a bad transistor or other components up to the IC itself. This does not look good (now I know why it was in the surplus bin).
- Set scope to fastest TIME/DIV setting, a very faint signal was seen. The STROBE pulse width was extremely narrow, but it was there and at regular intervals! This last wire is BROWN.

Hardware - Microprocessor

- Connected it to the Z80, wrote some machine language code and entered it into the HEX keypad (yes, it was load your own code by the numbers).
- It turns out the STROBE signal was too narrow for the Z80 clock and the program captured different data on every pass.
- Solution? A 555 timer.

Hardware – One Shot

- One-shot pulse widener (555 timer)
- Many descriptions of 555 timer. One example is http://www.sentex.ca/~mec1995/gadgets/555/555.html
- Soldered it together (no breadboard).
- Data was consistent with repeated scans.
- The hardware is working !

Encoding Standards

- From previous readings I knew that anything involving electrical signals will have a 'Standard', like IEEE or ISO.
- Fortunately, Toronto had a large library with a research archive just north of Young and Bloor.
- When I asked if they had any IEEE or other standards for magnetic card readers. She blinked once, reached behinder her, without looking, and pulled it off a shelf.
- I told her I was impressed that she knew where it was and she said the man in front of me had JUST returned it.

Encoding Standards

- This (now outdated) ISO standard described the physical card, where the tape is located, how many tracks of data there are (three, but not on all cards) and most importantly, the bit encoding of each track of data!
- For this presentation I could not find that standard as it has been superceded. A link to the updated version is in the references but costs $$$ to get.

Encoding Standards

- Re-wrote the Z80 code to shift in the data bits, 5 bits per character, and store it more readable.
- This reader only reads track two for financial institutions.
- Side note: swiping the card backwards is possible. First and last characters can be used to distiguish direction so data can be flipped.

Demo

- Arduino Uno
- Magnetic stripe card reader
- USB powered from laptop
- Sketch will simply copy the data bits to the serial port.

References

1)History of Magnetic Stripe Cards: http://spectrum.ieee.org/computing/hardware/the-long-life-and-imminent-death-of-the-magstripe-card

2)Encoding layout: https://en.wikipedia.org/wiki/Magnetic_stripe_card

3)ISO Standard: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=61936