



# Dactylize: Automatically Collecting Piano Fingering Data from Performance



David A. Randolph and Barbara Di Eugenio

University of Illinois at Chicago  
Department of Computer Science  
drando2@uic.edu

## 1. Overview

### Dactylize:

- Collect piano fingering data directly from performance
- Develop accurate fingering corpora economically
- Based on preliminary evaluation, reach accuracy over 99% at rates up to 12.5 notes per second



Figure 1: Monitor output from demonstration system.

### Motivation:

- Extensive corpora essential for developing cognitive and computational models
- No significant piano fingering datasets available
- Existing models inadequately evaluated
- Manual annotation expensive

### Approach:

- overlay each key of MIDI-enabled keyboard with foil
- Attach conductive tape to pianist's fingers
- Wire all to patch bay circuit
- Key wire, key foil, finger tape, and finger wire = patch cord
- Synchronize with MIDI events to determine fingerings

**Cost:** About \$300 plus cost of underlying digital piano.

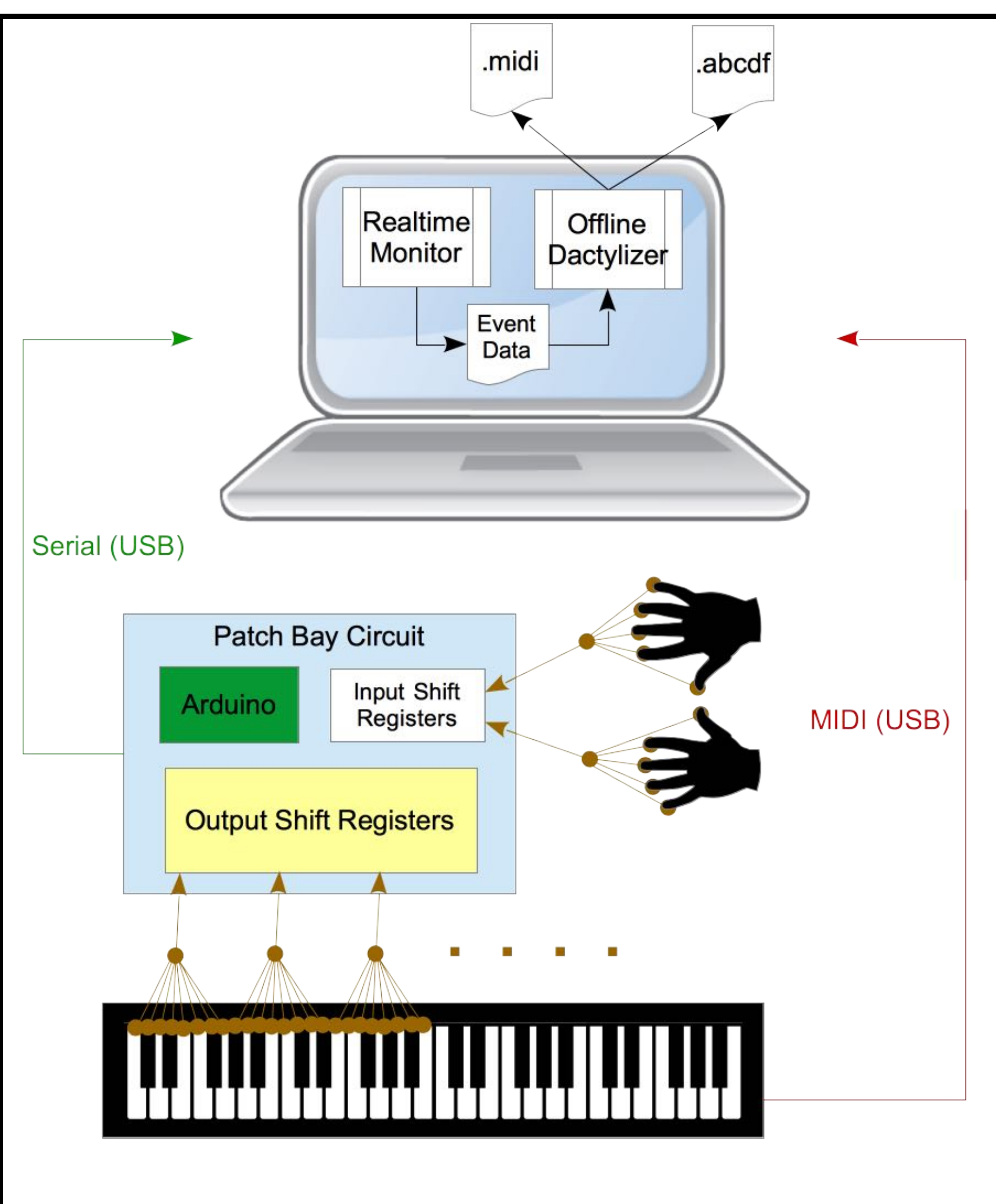


Figure 2: Dactylize architecture. Credits: Piano by Juan Pablo Bravo and hand by Dmitry Baranovskiy, both from the Noun Project.

## 2. Key Wiring

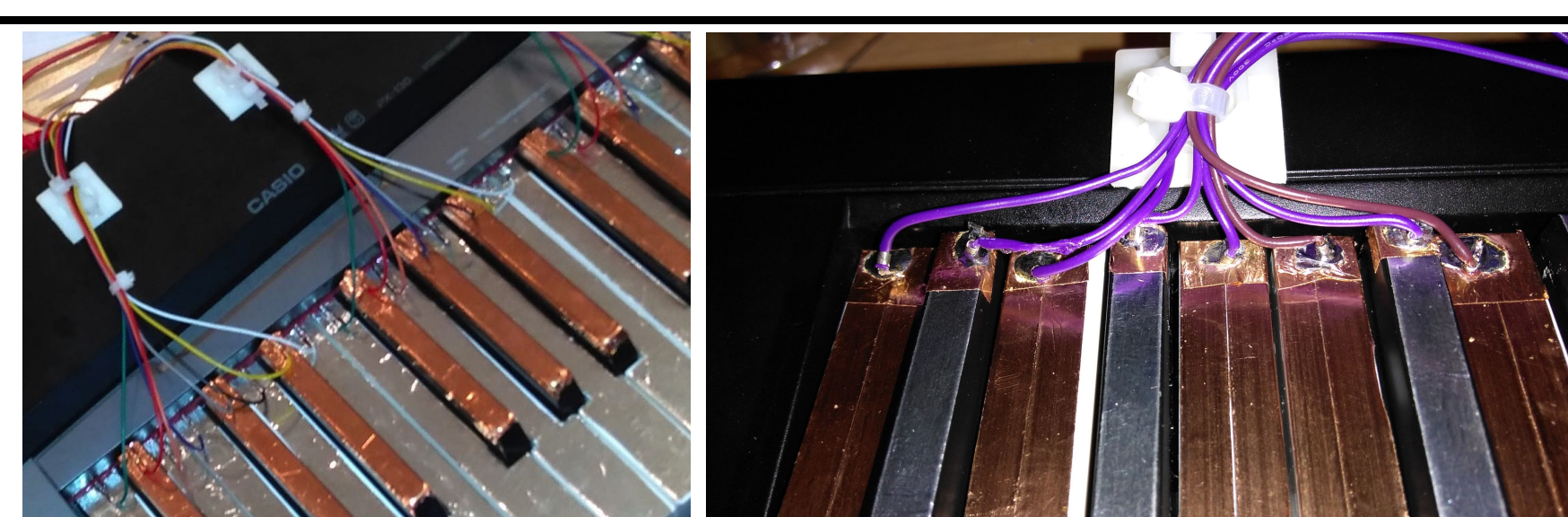


Figure 3: Wired key foil overlays on production system with mechanical (adhesive) connections (left) and demonstration system with soldered connections (right).

## 3. Finger Wiring

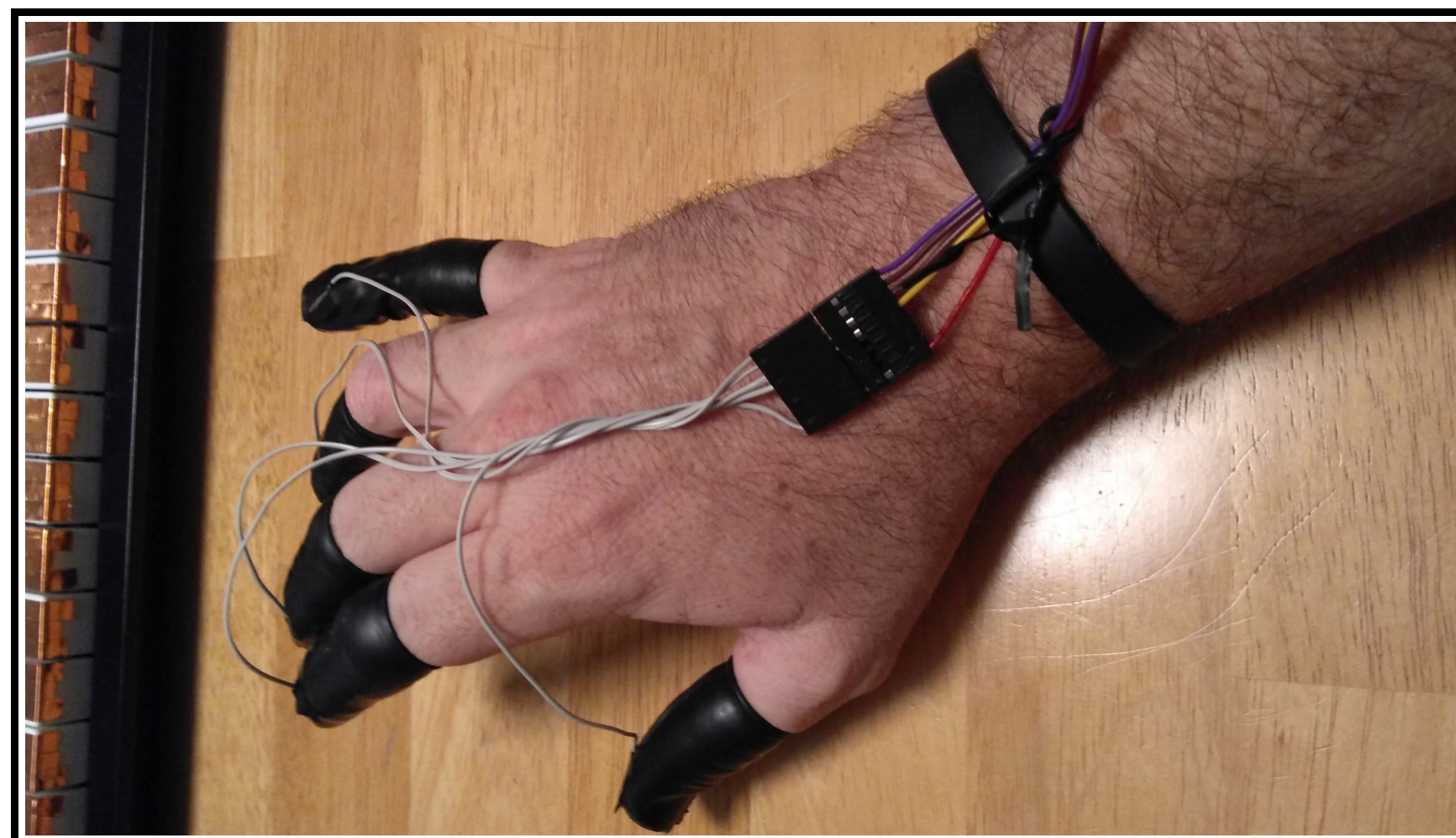


Figure 4: Finger attachments.

## 4. Micro-Controller (Arduino) Circuit

Adapted from Drymonitis [1], circuit designed for 88 keys currently supports 79.

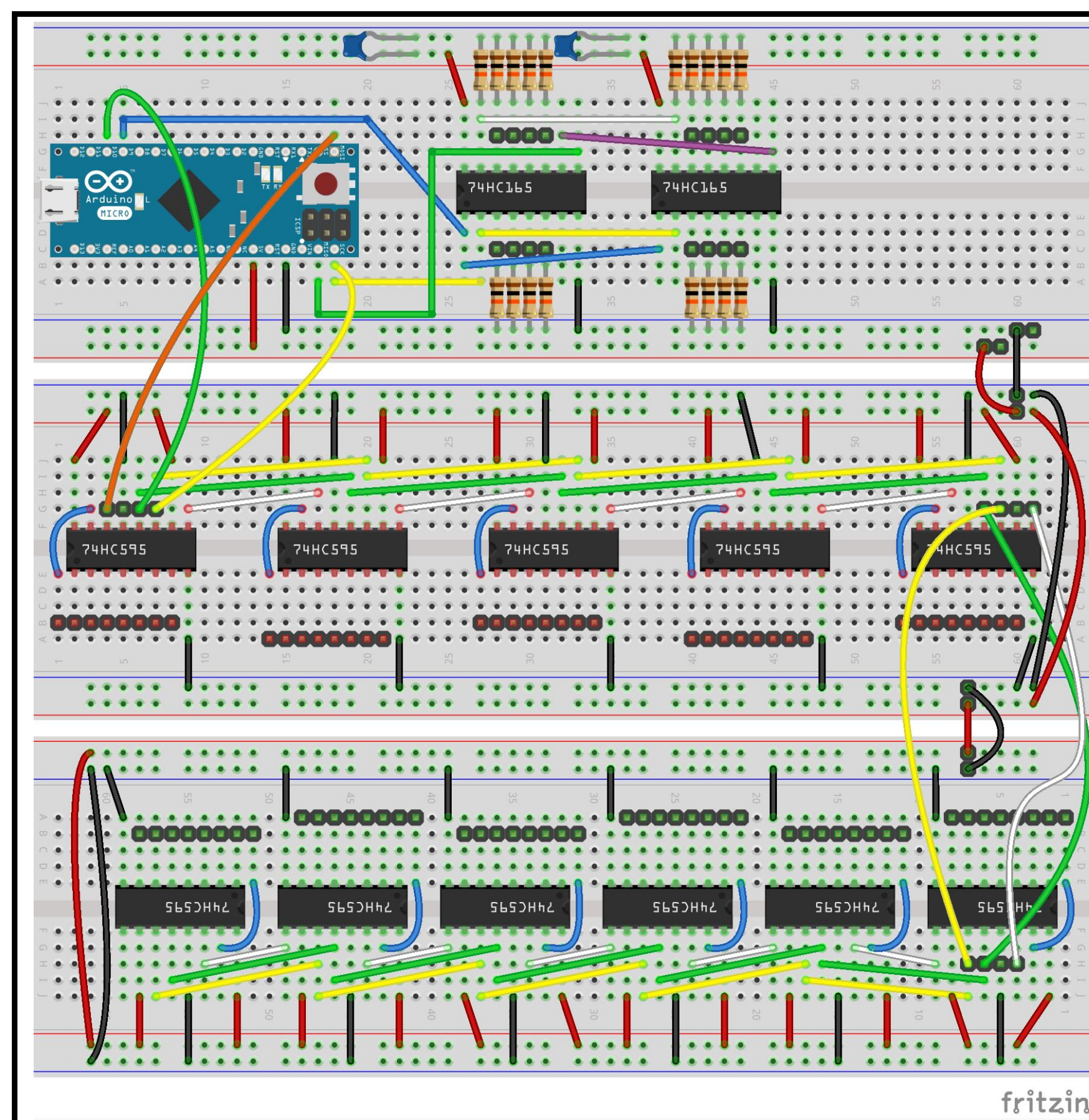


Figure 5: Fritzing diagram of complete patch-bay circuit.

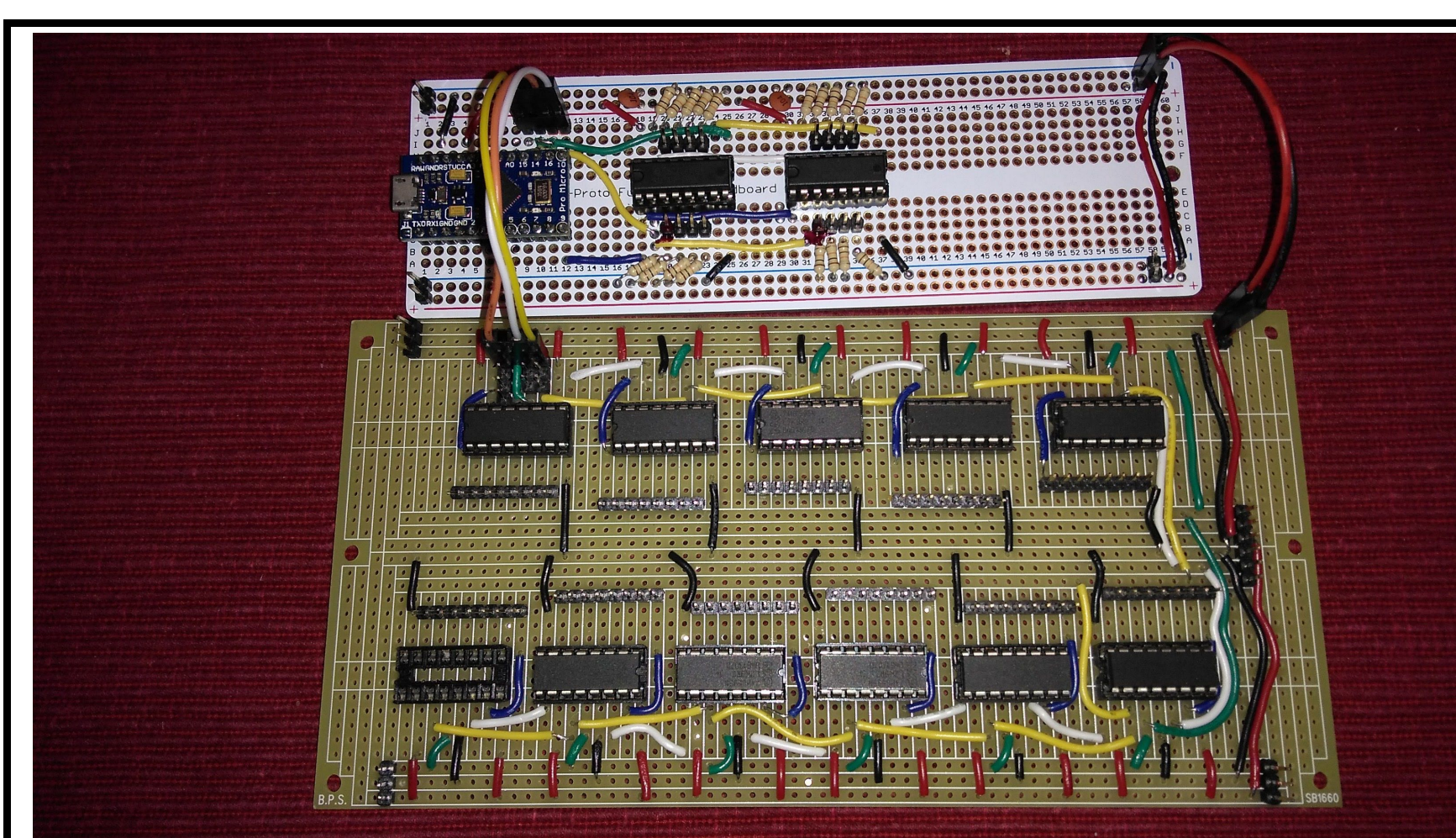


Figure 6: Current micro-controller circuit implementation.

## 5. Software

### Micro-controller code runs on Arduino:

- Based largely on Drymonitis's implementation
- Polls circuit for connection state changes
- Serial output to computer maps fingers to keys

### Monitor code runs on computer during performance:

- Multithreaded Python script
- *serial* and *mido* modules
- Microsecond time stamp serial and MIDI events
- Output to separate files to avoid contention
- On completion, call offline "Dactylizer" Perl script

### Dactylizer code runs offline:

- For each recorded MIDI note
  - Set "striking" finger as last one to contact note prior to onset time
  - Set "releasing" finger as last one to touch key prior to note's ending time, or, if no such finger detected, set as striking finger<sup>†</sup>
- Output machine readable MIDI and abcDF [2]

All code is open source and is maintained on GitHub at <https://github.com/dvdrndlp/dactylize>.

<sup>†</sup>If a release event is the last one detected for a key prior to the note onset, the algorithm will accept the first contact event prior to the note's ending time.

## 6. Demonstration System

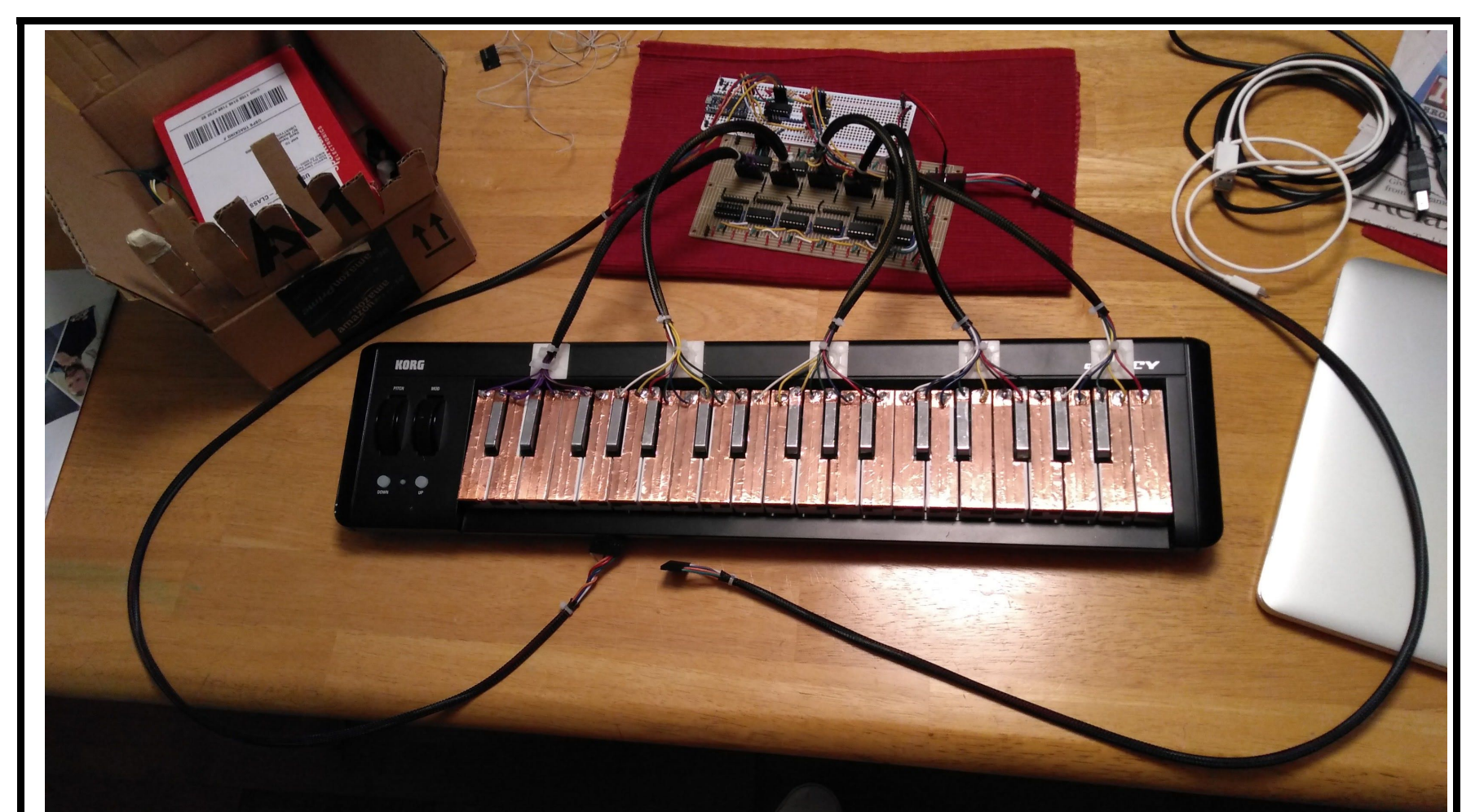


Figure 7: Demonstration Korg microKEY system.

## 7. Production System

Nearing completion.



Figure 8: Production Casio PX-130 keyboard wiring.

## 8. Preliminary Evaluation

- First author (non-pianist) "performs"
- Demo system: 0.75-inch wide keys
- Simple monophonic scale-like passage on white keys
- Right hand only
- 17 25-note trials
- Tempo increase 4.2 to 12.5 notes per second
- 421 of 425 notes (99.0%) correctly fingered
- 0 notes incorrectly fingered

## 9. Acknowledgments

Thanks to Alexander Demos, David Meyer, and Alex Radosavljevic. This work has been supported by a Provost's Award from the University of Illinois at Chicago.

## References

- [1] Alexandros Drymonitis. A Patch-Bay Matrix Synthesizer. In *Digital Electronics for Musicians: Build Intuitive Electronic and Electroacoustic Music Interfaces*, chapter 10, pages 417–480. Apress, New York, 2015.
- [2] David A. Randolph and Barbara Di Eugenio. Easy as abcDE: Piano fingering transcription online. In *Extended Abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference*, 2016 (in press).