Single-handed wrist-mounted keyboard

Peter Norstein
Department of Informatics
University of Oslo
Oslo, Norway
peterno@ifi.uio.no

***Abstract*—Keyboards are normally bound to a flat surface. Being able to use a keyboard without having a surface available would be valuable for mobile applications. We present a simple approach to create a limited wrist-mounted keyboard with modifiable keys.**

I. Introduction

KEYBOARDS are used every day to write text on computers. Being able to use a keyboard even if you only have the use of one hand would be valuable.

Building a standard shape and size keyboard is quite easily done by having a Printed Circuit Board made to order and soldering on the keyboard switches. This approach leaves much of the work to others. With minimal outside help, you can create a custom keyboard and mount it to your wrist. This enables the keyboard to be programmed to do many different things if you choose.

Using modifiable keys, a single 16-input keyboard is able to replicate a full textual keyboard. An ALT-key switches between the left and right-hand side of a keyboard (figure 1). Although more cumbersome than a regular keyboard, this would make it possible to write with a single hand.

Other uses for this keyboard is as a hotkey-board. Each key could start a sequence of actions defined by the programmer to simplify regular tasks. Examples include progressing a slideshow, starting a program which runs a routine for a robotic appliance or controlling video playback. Joe Coburn's blog-post [1] shows how to create a hotkey-board. My implementation is heavily inspired by this post.

II. The problem

Using a keyboard on a table is a restricting experience if you want to have your limbs separated. With a regular keyboard it is cumbersome to use without a stable flat surface.

As technology progresses, it is natural that devices we use move into broader domains than before. Connecting a mobile phone to a keyboard might become a regular occurrence in the future. Different approaches to input devices are needed for those uses.

III. My Idea

Creating a small keyboard with straps to fasten it to a users hand will enable the user to write without having a dedicated surface.

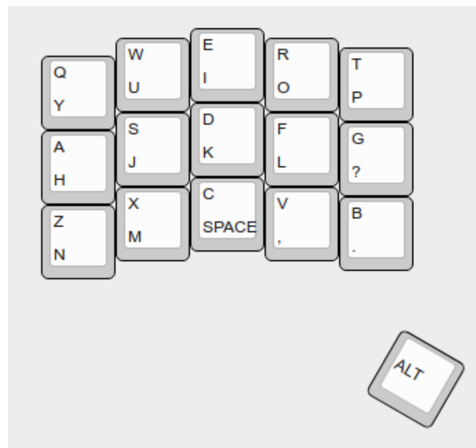

Fig. 1. Example of putting all English-alphabet characters on a single 16-input keyboard. Lower characters are input by holding ALT when pressed.

The keyboard is made by using key-switches (Cherry MX Black [2]), an Arduino Micro [3] and a laser-cut board of 1.5mm acrylic. The design is created with Keyboard Layout Editor (KLE) [4] and exported to DXF-format by copying the "Raw Data" from KLE into Plate & Case Builder [**?**]. We create a casing for the keyboard using MakerCase [5] by specifying the dimensions of our box and the acrylic material. Using SAI FlexiDESIGNER [6] we import the DXF-formatted files to combine and refine the design. We cut the board on an Epilog Zing 30 Watt laser-cutter [] according to the design.

The key-switches are put into their laser-cut (14mmx14mm) slots and soldered to an input on the Arduino and ground as shown in figure 2.

A strap is needed to fasten the keyboard to your hand. The holes for the straps are cut using the Zing laser and their position relies heavily on the users physique. We used a zip-tie to fasten it, which works moderately well. A Velcro strap is a more suitable solution, which is not featured in the figures here.

Figure 4 shows how the keyboard is fastened to the back of your hand with a strap. An optional wrist-strap can be used for more comfortable operation of the keyboard.

Using the PULLUP functionality of the arduino, the switches need only connect the pin from the arduino to ground to signal that the key has been pressed. Figure 2 shows the schematic for soldering the switches to the Arduinos pins. Figure 3 shows the completed soldering board with 17

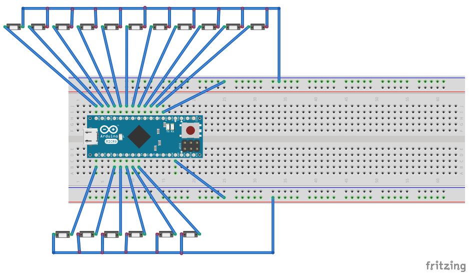

Fig. 2. Schematic of how to solder the key-switches to the Arduino. The keys are soldered to ground and a single input on the Arduino.

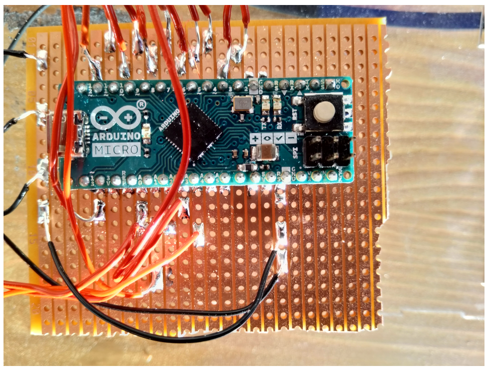

Fig. 3. Physical soldering of the key-switches to the Arduino. The keys are soldered to ground (black wires) and a single input (red wires) on the Arduino.

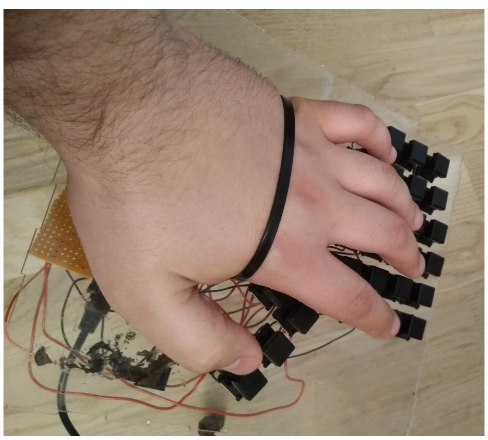

Fig. 4. Strap fastening hand to keyboard

connected keyboard switches.

The completed keyboard-prototype has too many keys as shown in figure 5 and 6. Use of all keys was very difficult when testing the layout in practice. A more robust and tailored harness would be required to be able to reach all these keys. Having the fastening mechanism on the underarm instead of the wrist might be able to achieve this. This would make the whole mechanism more stiff, which would enable the user to stretch further without moving the keyboard around. Having this many keys would require a slight reworking of the logic to enable the Arduino to have enough inputs.

The full Arduino code is given in the Appendix.

Using an Arduino Micro (with the ATmega32u4) it is possible to send keystrokes or mouse movement to a computer with

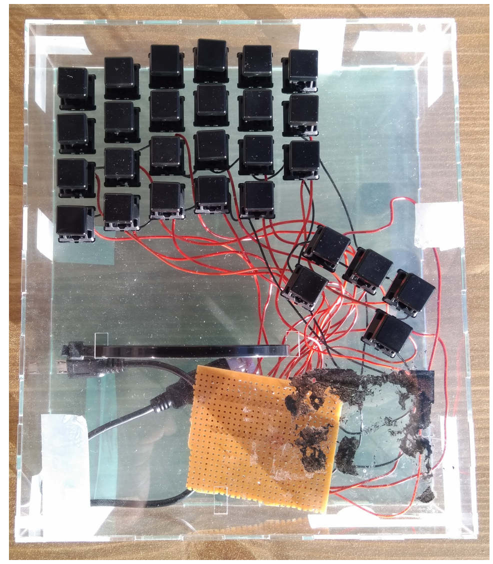

Fig. 5. Frontside of the completed keyboard-prototype.

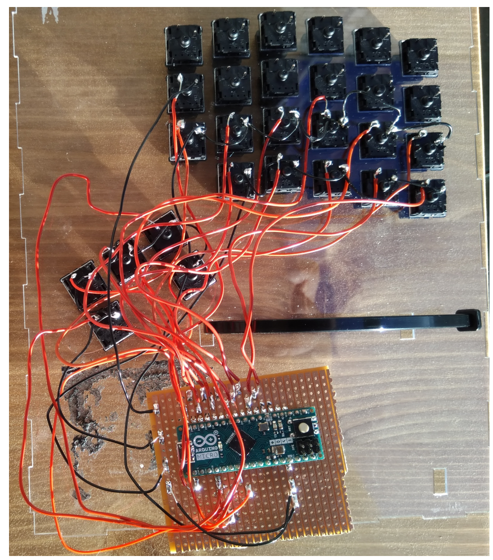

Fig. 6. Backside of the completed keyboard-prototype.

the Keyboard [7] and Mouse [8] functions. When checking inputs on the pins, the code can be modified to do whatever a keyboard or mouse is able to do. In our implementation **??** we use most of the buttons as simple keys, with modifiers to enable the use of all conventional keyboard buttons. Modifying the code to launch applications, write pre-written sentences or more complex actions is as simple as adding some lines in the Arduino-code.

Making these changes requires an Arduino IDE.

IV. Related Work

Little has been published as scientific papers relating to wrist-mounted keyboards. The closest related work is a prototype glove (GAUNTLET) for inputting the most common characters in the English language made by a group of students at the University of Alabama in 2012[9]. The device uses bluetooth and a capacitive input device to send characters based on which part of the hand the thumb touches. They did not publish any scientific papers on the topic.

The company Tap Strap Inc. [10] has created a hand-mounted device for inputting keys and mouse gestures. It is a high-tech strap equipped with accelerometers to sense when the user taps. This device requires an improvised surface.

The company PCDMaltron makes ergonomic Single Handed Keyboards for use by people without the use of both limbs. These keyboards are large, specially created keyboards for use on a desktop computer. They offer a different key-order, which does not follow the QWERTY-standard to enable the best use of a single hand for typing.

V. Conclusions and further work

We have shown that it is possible to create a simple wrist-mounted keyboard with modifiable keys using simple components. Making a functional wrist-mounted keyboard is not extraordinarily complicated. In conclusion the keyboard is a moderate success, as the number of keys is limited, and the reach of the authors hand is not quite good enough with the current design.

Creating a wrist-mounted keyboard that is ergonomically suitable is a much more difficult task. Adding a spherical structure to the middle of the housing to wrap your hand around would help lessen the strain from extending your fingers.

Further work includes adding a battery and connecting a bluetooth communication module. This would enable the keyboard to work wirelessly, which would make it much more useful.

Adding a right-handed wrist-mounted keyboard would enable the user to write more efficiently. Having this wrist-mounted would make it difficult to use a mouse intermittently. Therefore you could add a mouse tracking-ball to the right-hand keyboard to enable mouse control. A different approach would be to harvest a laser-pointer mouse to enable the right hand keyboard to control mouse movement.

Appendix

Source code for Arduino

```
#include <Keyboard.h>

#define NUM_KEYS 17
#define MODIFIER_KEYS 2

int keys[] =
{2, 3, 4, 5, 6, /* Regular keys */
 7, 8, 9, 10, 11, /* Regular keys */
 12, 13, A0, A1, A2, /* Regular keys */
 A3, A4};/* Modifier keys */
bool shift = false;
char keys_to_press[] =
{'z','x','c','v','b', /* First line */
 'a','s','d','f','g', /* Second line */
 'q','w','e','r','t', /* Third line */
 'n','m',' ',',','.', /* ALT First */
 'h','j','k','l','?', /* ALT Second */
 'y','u','i','o','p'}; /* ALT Third */

void setup() {
  Keyboard.begin(); // setup keyboard
  for (int i = 0; i < NUM_KEYS; ++i) {
    // initilize pins
    pinMode(keys[i], INPUT_PULLUP);
  }
}

void loop() {
  //Don't check modifier keys
  for (int i = 0; i < NUM_KEYS -
      MODIFIER_KEYS; ++i) {
    // check buttons
    if (readButton(keys[i])) {
      doAction(i);
    }
  }
}

boolean readButton(int pin) {
  // check and debounce buttons
  if (digitalRead(pin) == LOW) {
    delay(10);
    if (digitalRead(pin) == LOW) {
      return true;
    }
  }
  return false;
}

void doAction(int action_val) {
  char key_to_press;
  shift = false;
  //A3 is the ALT key. Check if it is
      pressed.
  if (digitalRead(A3) == LOW) {
      shift = true;

  }
  //Use the alternative inputs if shift
      is true
  int shift_val = shift ? 15 : 0;
  // perform tasks
  key_to_press = keys_to_press[action_val
      +shift_val];

  Keyboard.write(key_to_press);
  delay(100);
  //Keyboard.release(key_to_press);
}
```

Acknowledgment

The author would like to thank the ROBIN research group, especially Mats Erling Hvin, Yngve Hafting and Vegard Syseth for all the help they have given.

REFERENCES

[1] J. Coburn. (2016, Nov.) Make your own custom shortcut buttons with an arduino. [Accessed May 2019]. [Online]. Available: https://www.makeuseof.com/tag/make-custom-shortcut-buttons-arduino/
[2] Cherry. Cherry mx black keyboard switches (bought at rs components). [Accessed May 2019]. [Online]. Available: https://no.rs-online.com/web/p/keyboard-switches/0664569/
[3] Arduino.cc. Arduino usb-mouse reference. [Accessed May 2019]. [Online]. Available: https://store.arduino.cc/arduino-micro
[4] I. Prest. Keyboard layout editor. [Accessed May 2019]. [Online]. Available: http://www.keyboard-layout-editor.com/
[5] J. Hollander. Makercase. [Accessed May 2019]. [Online]. Available: https://www.makercase.com
[6] SAI. Sai flexidesigner. [Accessed May 2019]. [Online]. Available: https://www.uscutter.com/FlexiDESIGNER-Cloud
[7] Arduino.cc. Arduino usb-keyboard reference. [Accessed May 2019]. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/usb/keyboard/
[8] ——. Arduino usb-mouse reference. [Accessed May 2019]. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/usb/mouse/
[9] J. Liu, S. Doud, D. Kirby, and C. Heath. News story about the gauntlet keyboard. [Accessed May 2019]. [Online]. Available: https://www.uah.edu/news/research/uahuntsville-students-hope-glove-keyboard-will-revolutionize-use-of-devices-with-one-hand
[10] T. S. Inc. Tap strap keyboard. [Accessed May 2019]. [Online]. Available: https://www.tapwithus.com/