PC Buttonpad

Team 12 – Yicong Dong, Stephanie Jaster, and Adit Umakanth
ECE 445 Project Proposal – Fall 2019
TA: Mengze Sha

1 Introduction

1.1 Objective

Keyboards and mice work well with simple tasks, but are not the quickest way to interact with computers. There are some actions people perform on a regular basis that take many mouse clicks or keyboard button presses, which are not only annoying to do but also tiring and time consuming, but could be condensed down into one simple press of a button. Such actions include but are not limited to opening up certain pages altogether in a browser, quickly adjusting volume and brightness of the PC, taking screenshots and sharing to social media with one click, etc. In order to increase productivity, decrease the repetitiveness of working on a computer, and enhance the experience and fluidity of working on a computer, we believe it is necessary to engineer more user-friendly hardware designs of the mentioned characteristics.

We therefore propose to make a compact and portable unit with sensors, buttons and LEDs that can be connected and used on the three major operating systems: Windows, macOS, and most distributions of Linux. These buttons will perform whatever repeated actions the user assigns it, and the pad does not need other power supply except the output from the Universal Serial Bus (USB). Sensors and LEDs shall provide more advanced features that further assist the user with work.

1.2 Background

Macro buttons are not new and exist on some high-end keyboards and specialized keypads on the market [1]. Workflow of multiple actions are also existing features of many operating systems. One of our team members has experience using the stock Automator of macOS, and it is still only on the software side with no hardware dedicated to it to make it easily accessible. In addition, the number of functions are limited and mostly only on the stock application.

Our project is different because it promises to let the user customize the button functionality to their wishes instead of relying on forced presets, spend a much lower cost, and still able to keep their favorite keyboards on the desk while having our solution in parallel with the existing.

1.3 High-level Requirements List

- Buttonpad must be compatible with the mainstream PC operating systems: Windows, macOS, and popular distributions of Linux.
- Individual button functions should be easily customizable by an average computer user, without requiring any programming or technical knowledge, and buttonpads must be durable, i.e. do not fail when used on their daily basis.
- Buttonpad should be portable and small enough to fit in a reasonably sized backpack.

2 Design

2.1 Block Diagram

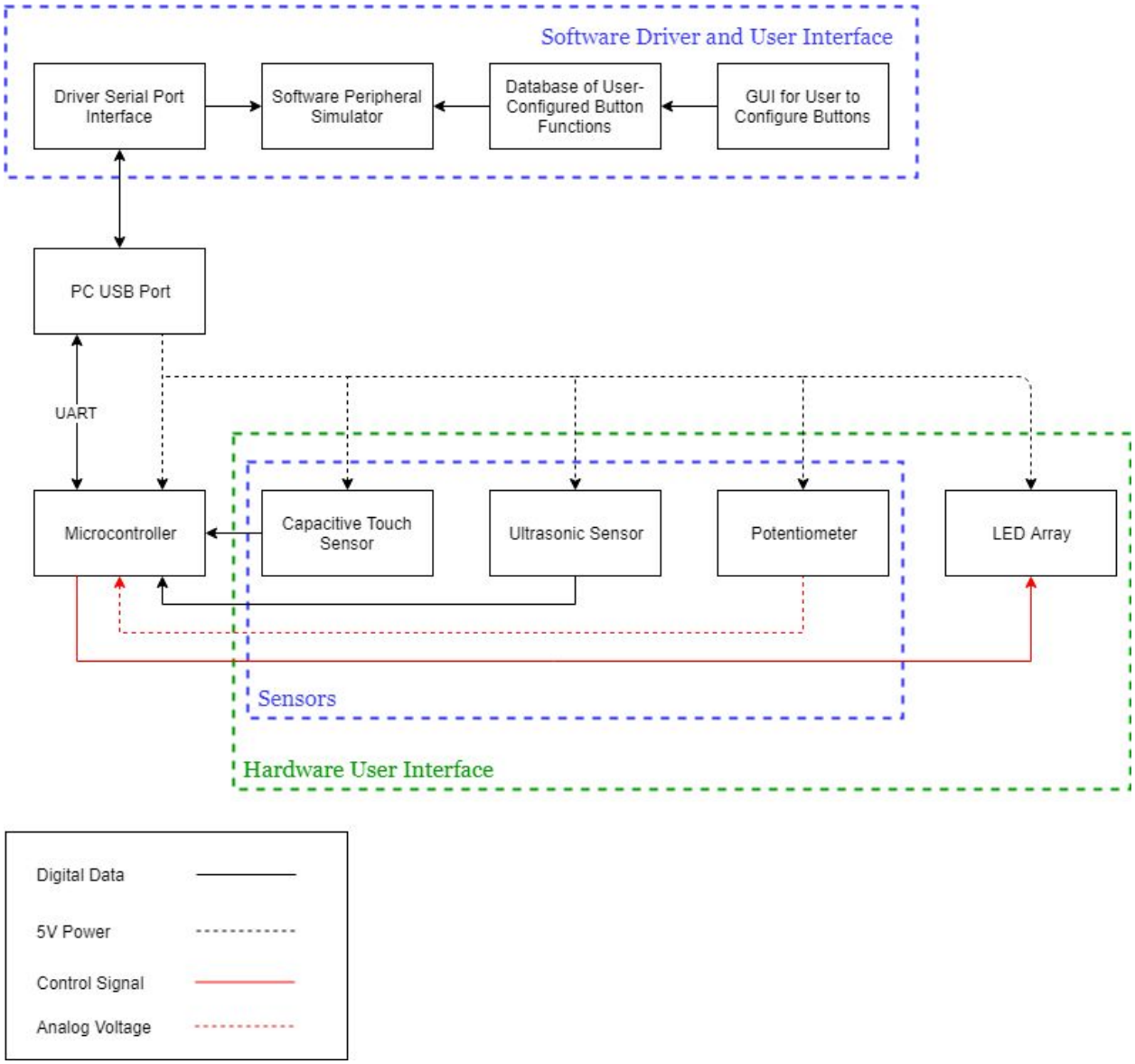

Figure 1. Block Diagram of PC Buttonpad

As illustrated in Figure 1, capacitive touch sensor detects button presses and sends a signal to the connected computer through a microprocessor and USB chip. The connected computer has drivers installed which decodes these signals and performs the user-assigned actions. The USB port of the user's computer is expected to power the whole circuit. Additional sensors serve as part of the user interface and provide extended features and functionalities.

2.2 Physical Design

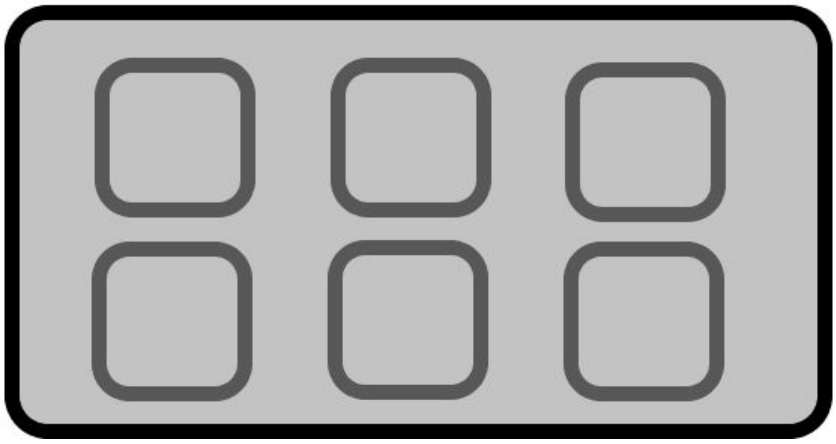

Figure 2. Physical Design

This device will ideally be about 10 cm wide. However, this size specification and number of buttons could change during the further development of this project, depending on factors like ideal number of buttons and size constraints introduced by the hardware.

2.3 Functional Overview

2.3.1 Software Driver and User Interface

Driver Serial Port Interface: This is the piece of software that allows the microprocessor to communicate with the user's computer. A press of the button or trigger of ultrasonic sensor is passed on to the microprocessor and sent to the USB Serial Port of the PC, which is being listened to by this piece of software. It carries out the relevant actions depending on the signals from the hardware.

Software Peripheral Simulator: Instead of having the hardware mimic the signals of a keyboard or mouse, we use the serial port interface to read a custom signal, and then use this Software Peripheral Simulator module to simulate the click of a mouse or typing on a keyboard. This offloads all the limitations of storing information for each button onto the computer instead of our hardware. It also allows further customizations beyond keyboard and mouse interactions such as running shell commands or specific libraries for certain software if required. We will use an open-source library for this module instead of creating this functionality from scratch.

Database of User-Configured Button Functions: This is where are all saved keyboard and mouse shortcuts the user sets are stored. Whenever a signal is received through the serial port, this database is checked to find out what actually needs to be executed by the software peripheral simulator. The user does not have direct access to this database, but can freely and easily make changes to it through the GUI for User to Configure Buttons.

GUI for User to Configure Buttons: This module is the core of customizing the buttons to the user's liking. An easy to use GUI will allow the user to choose specific buttons and assign desired keyboard shortcuts or command line executions.

2.3.2 Control System

Since our project is a blend of hardware and software, the main control system will be the Microcontroller on the hardware side and the driver program on the software side. These two modules communicate with each other through the USB Serial Port and a Universal Asynchronous Receiver-Transmitter (UART) to convey data like battery levels of the connected computer and the buttons pressed.

2.3.3 Sensors as User Interface

Capacitive Touch Sensor: Since we do not need high resolution for each single button, it is best to build capacitive touch sensors using surface capacitance, where "only one side of the insulator is coated with conductive material. A small voltage is applied to this layer, resulting in a uniform electrostatic field. When a conductor, such as a human finger, touches the uncoated surface, a capacitor is dynamically formed" [2]. We would build nine of these sensors and place them in a 2×3 grid-fashion on the buttonpad. Capacitive touch sensors ensures that there is no mechanical parts in the buttons, making the design simpler and more durable. Choice of material will need further experiments, but most likely will be what is already available in the ECE Machine Shop.

Ultrasonic Sensor: This sensor really comes down to measuring how far an object is. Using the time it takes for sound to travel to the object and reflect back to get received, it can find the distance from the speed of sound through a simple multiplication. In our case, we would allow a certain range of distance that the user can wave his/her hand passing by two ultrasonic sensors, and this shall allow the controller to figure out if the movement is to the left or to the right. With this movement, the profile of the pad switches and other functionalities can be activated. These sensors will be properly arranged such that using the buttons on the pad would not trigger them mistakenly.

Potentiometer: The potentiometer uses voltage division law, and can have varying voltage readings if the user turns the knob. We shall utilize this feature by measuring the change in voltage reading due to the user turning it, and reflect that change as change in volume or brightness depending on the profile that is active on the PC. We will use a potentiometer with as many rounds as possible such that if the user keeps turning in one direction, the potentiometer would at least be functional for a long time period. On a weekly basis, the user should be reminded to calibrate the potentiometer to the center position by reading the absolute voltage (instead of the change in voltage).

2.3.4 Additional Hardware User Interface

In addition to sensors mentioned above as part of the user interface, we could also include an LED array to indicate the battery level of the PC. This can be achieved with a binary decoder that receives signals from the PC. As for the LED array, we may use four or five LEDs of the same color, and linearly indicate

the battery percentage value to the closest proportional ceiling. For example, for four LEDs, 60% battery level would have three LEDs on and one off as it is in between 50% and 75%.

2.4 Block Requirements

2.4.1 Software Driver and User Interface

Driver Serial Port Interface: This module must have a fixed protocol and be able to distinguish between enough signals such that the exact button that has been pressed can be encoded through the microprocessor.

Software Peripheral Simulator: This module must be able to simulate any keyboard button press.

Database of User-Configured Button Functions: This database must be designed in such a way that it can hold a reasonable number of steps of actions for each button press (eg. Open Chrome, Type https://youtube.com into address bar with one button press).

GUI for User to Configure Buttons: An average computer user should be able to configure their own custom shortcuts with only the intuitiveness of the on-screen buttons, tooltips, and a help box.

2.4.2 Control System

The microcontroller must be able to send commands and receive commands from the driver serial port interface through the UART.

2.4.3 Sensors as User Interface

Capacitive Touch Sensor: The idle power consumption of the sensor of choice used must be reasonable. The sensors must be triggered if the user physically touched them, and shall not trigger if the user placed his/her finger(s) $\geq$1 mm away from the buttons. Buttons must be triggered one at a time.

Ultrasonic Sensor: The idle power consumption of the sensor of choice must be reasonable. The sensors must allow the user not to touch them to switch profile. They must not be triggered if the user is using the capacitive touch buttons. In that case, it is best to arrange it at the farther side of the pad and set a trigger range from one inch to one foot. When picking the ultrasonic sensors, we must make sure to spend less than one third of the total budget.

Potentiometer: The resistance must be large enough such that the total power consumption of the potentiometer circuitry does not exceed 2.5 mW. Reminders for calibrating the potentiometer must be sent to the user through the software interface at least on a weekly basis.

2.4.4 Additional Hardware User Interface

The LEDs must have a lifetime close to the general 50,000 hours expected [3]. We will not use red LEDs because they might look annoying to some people. The refresh time will be as short as half a minute such that the correct battery level is displayed promptly.

2.5 Risk Analysis

One major risk of this project is the choice of material for our capacitive sensors. Different materials have different conductivities as well as dielectric constant, and will indefinitely affect the capacitance before and after the touch. In order to ensure the sensitivity and reliability of the buttons, we must have testing done as soon as possible.

3 Ethics and Safety

Abiding to IEEE ethics, we are responsible for implementing and maintaining professional and ethical standards by which we follow throughout our design process and afterwards. As a team, we will ensure that all data and communications are truthful, following #3 of the IEEE Code of Ethics [4]. This includes communications between each other, professors, TAs, classmates, and future users. To create a safe working environment for our team and future users, we are aware of our individual technical skillset and will not take on technical tasks in which we are not qualified to complete. Assistance will be required in these cases to avoid any unsafe practices that would go against #6 of the IEEE Code of Ethics [4].

Our team values our compliance with the IEEE Code of Ethics through honesty, safety, and our technical work output. We will uphold #7 of the IEEE Code of Ethics to stay true to each other and our individual accountability and contributions within the project [4]. We are each responsible for our individual work and will face proper consequences for discrediting another's ideas or causing extreme setback for our team's progression.

According to #1 and #2 of the IEEE Code of Ethics, we agree to be transparent with any party affected by our work whether it regards safety, our design, or conflicts of interest [4]. It is with utmost importance that we are honest and straightforward with all affected parties, especially when it comes to safety as that is our highest priority in this process. Clarity and guidance will result if a party misunderstands the makeup of our design, technical and physical, the purpose of our product or its safety concerns.

We are also in agreement with the ACM Code of Ethics and Professional Conduct. While this code similarly reflects that of the IEEE Code of Ethics, it points out leadership responsibilities that we as a team must inherit. Our top priority is to ensure the safety and wellbeing of all users as listed in #3.1 of the ACM Code of Ethics and Professional Conduct [5]. As leaders, we take complete ownership of our product and strive towards creating a safe and educational environment.

Our display of commitment to the ethical and professional standards outlined will continue during and after our project design and implementation.

The project proposed contains safety hazards that all users must be aware of. The user interface includes several sensors which should be kept dry at all times. Moisture and other types of liquid have the potential to leak into the interior of the PC Buttonpad and could cause damage to the circuitry inside. Corrosion, mechanical failure, electrical shortages, or the creation of fire may occur in this instance. If any liquid leak is suspected, immediately shut off your PC and unplug the device from your PC. To prevent this from happening, we plan on creating a physical model that will seal any gaps between the interior circuitry and exterior of the device.

The buttons are made up of capacitive touch sensors which can malfunction if pressed with too much pressure. Excessive forces placed upon these sensors can result in damages, disconnections, and loss in overall functionality of this product. To prevent these instances from occurring, the buttons are placed a slightly lower level compared to the rest of the project surface. However, this does not completely avoid the damaging of our product through forcible touch. In the case that buttons do get damaged, do not disassemble the device, as one incorrect wire placement could lead to overheating elements.

Overheating can also occur if 5V of power is not provided. Each module of our device is designed around a 5V power input from the USB, so any more or less voltage could affect the current flowing through the system. Since the device relies on PC power, there is no worry that the voltage will exceed 5V. Monitors that supply a 3.3V USB port are labeled, so it is important for the user to plug the device into the correct port when connecting it to the PC monitor.

Our team has successfully completed the Lab Safety Training required for access to any lab. Following campus policy #RB-13, Campus Environmental Health and Safety, we take full responsibility for maintaining a healthy and safe environment for ourselves and the rest of the University of Illinois at Urbana-Champaign community [6].

4 References

[1] "Keyboard with Macro Keys," *Amazon*, 2019. [Online]. Available: https://www.amazon.com/slp/keyboard-with-macro-keys/gnyyx93vhu2484f. [Accessed Sep. 20, 2019]

[2] "Capacitive sensing," *Wikipedia*, Aug. 16, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Capacitive_sensing. [Accessed Sep. 16, 2019]

[3] "LED Life Expectancy," *Electronics Weekly*, 2019. [Online]. Available: https://www.electronicsweekly.com/blogs/led-luminaries/led-life-expectancy-2009-02/. [Accessed Sep. 18, 2019]

[4] "7.8 IEEE Code of Ethics," *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed Sep. 15, 2019]

[5] "ACM Code of Ethics and Professional Conduct," *Code of Ethics*. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed Sep. 15, 2019]

[6] "Campus Environmental Health and Safety," *Campus Administrative Manual.* [Online]. Available: https://cam.illinois.edu/policies/rp-13/. [Accessed Sep. 19, 2019]