### Problem 3

**Morgan Bacik, Ben Cooper, Patrick Johnson (Team 3)**

1. **Define the problem**
   1. **What is the problem? What are you looking for? What are you trying to accomplish? What’s the goal here?**

We are designing and developing a wearable assistive technology device. We will be using a microcontroller that runs using Python to create a device that can control a computer cursor with both left and right mouse clicks as well as have the WASD keys & Spacebar function, all without the use of individual fingers.

1. **Explore the problem and identify constraints.**
   1. **What defines control of a cursor and mouse clicks? What needs to be controlled?**

Control over direction and speed of mouse movement defines control of a cursor. The mouse clicks need to be able to respond to stimulus a single time (e.g. a single input yields a single output – one left or right click). For the cursor and mouse clicks, the mousepad needs to be controlled.

* 1. **What defines control of the WASD and Space keys? What needs to be controlled?**

Control of the WASD keys and Spacebar key would be in the form of having a single input on the microcontroller output the corresponding key on the computer. The keyboard needs to be controlled to accomplish this.

* 1. **Why do you need a microcontroller? What functionality does it provide? What are all the things it can do and which of those might be useful in your solution (and which are likely not useful)?**

The microcontroller provides the built-in sensors with the additional ability of being controlled using self-written code. Some of the things that this microcontroller can do are read acceleration, capacitive sensing, buttons, tap detection, photo sensing, neo pixels, on-board speaker, led lights, shake detection, switch, temperature sensing. Relevant sensors for the solution could be shake detection, switch, tap detection, acceleration, and capacitive sensing. Some likely not useful abilities would be led lights, on-board speaker, and photo sensing.

* 1. **Why do you need Python? What does it enable that the microcontroller alone can’t provide?**

Python is needed in order to control various factors and make customizations of the controller’s functionality within the code. We can customize the microcontroller and add features like light colors, delays, adjust the sensitivity, set different thresholds, etc. The microcontroller allows for the basic functionality and reception of things like movement, acceleration, sound, and touch, but using Python we can tell the controller what to look for and can alter the functionality adding various features, programmable tasks, and customization.

* 1. **What are the biomechanical/anatomic systems that might be relevant to achieving your goal? What do you need to know about them to utilize them effectively in your solution?**

We need to understand the anatomy of the hand and how it can be used to control the mouse and keyboard. Specifically, we need to know how to gain control *without* the use of individual fingers. This will require different forms of input such as moving the arm/hand in space or the user’s voice.

* 1. **What do you already know that is relevant to the problem?**

We know basic python commands to control the mouse and keyboard.

* 1. **What do you need to find out that is not given? (Hint: Consider why do you need this information before spending time getting it. Don’t waste time chasing down information you don’t need.)**

We need to find out how to control the device without the use of individual fingers. This might include finding more Python commands.

* 1. **What assumptions, if any, will you make to solve this problem. (Note: you are encouraged to make reasonable assumptions! However, you should ask your instructor about assumptions before proceeding with your solution.)**

We will assume that the person using this device has been informed and trained on its functionality prior to use. We will also assume that the microcontroller is functioning properly.

* 1. **Identify constraints on your solution, such as time, money, and materials.**

We are constrained by the abilities of the microcontroller. All we are given are the microcontroller and alligator clips. Time is also a constraint as we are given about a week to generate a solution.

* 1. **Identify criteria for success. What will constitute success? How will you know when you’ve achieved it?**

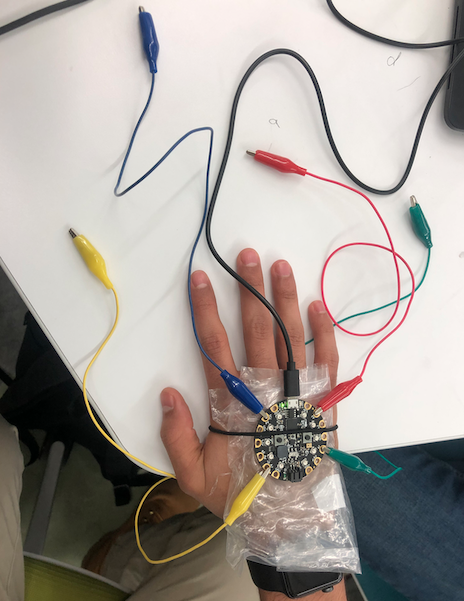
Success will be constituted when the microcontroller can successfully right click, left click, function as a mouse cursor, output letters WASD, and control the space key. If we can play a game which requires the use of those keys using the devices, it will constitute success. We will test each of the criteria individually to determine if it functions as desired. If all the requirements function successfully, we will have achieved success.

1. **Generate ideas. Brainstorm multiple possible solutions. Don’t immediately proceed with the first approach you think of—ideate and think through multiple different ideas before settling on the one you think most promising. Only then should you proceed to the next prompt.**
   1. **What are the possible solutions you can identify?**
   * Tilting the device and using the accelerometer input could be used to control the mouse. Capacitive touch pads can be used to control WASD keys and spacebar. Shaking the device could indicate a mouse click.
   * Use ‘keyboard.send’ code function to represent the WASD keys or SPACEBAR.
   * Attach alligator clips to the capacitive touch pads.
   * Have the acceleration of the device in different directions function as the movement of the mouse cursors.
   * Have the WASD keys work using the capacitive touch pads. Use the shaking function of the device count as a space bar whenever the device is shaken. Control right and left clicks by tilting the device far left for a left click and far right for a right click.
2. **Select the most promising solution and formulate a solution algorithm/design concept. Using the most promising idea identified following your ideation in #3, draft a series of steps that clearly bring you from where you are now to the end point you are seeking. If the algorithm/design involves programming, write and justify what the code will do but don't write the code itself (You will start doing that in #5).** 
   1. **What solution do you think most promising? Why?**
   * Right click – temperature sensor, because it can be a one-time click easily
   * Left click – photosensor, because it will be easy to detect a click as well as potentially adding a double click function and a click and drag function
   * Function as a mouse cursor – accelerometer, because these mimics the actual motion of a mouse and will be intuitive for the user
   * Output letters WASD - capacitive touch, this will be easy for the user because it is like clicking a key; the benefit of using the capacitive touch is that it can be used with any part of the body that changes the capacitance of the sensor
   * Control the space key – shake, because it can be easily implemented to print the space bar because as the controller senses shake, it will reference the keyboard and input a space click
   1. **What steps will be required to get you from your initial thinking about the problem to the solution you have chosen? Be detailed. Identify all of the step you can, but you do not need to perform those steps as part of this prompt, you will do so in response to the next prompt.**

The first thing we need to do is write a code:

* The code should consist initially of importing various libraries including math, time, keyboard, keycode, cpx, mouse, and any other necessary library
* Using the in-class assignments we are going to copy some of the codes and functions to get the microcircuit to function as we desire
* Using while loops, and if/else functions, we plan to use code to control when different actions occur. For example, if the acceleration value of the accelerometer is greater than a threshold value, we will have the mouse cursor move as a function of that acceleration.
* We want to take advantage of the capacitive touch pads by attaching alligator clips to them so that the user can function the WASD keys and space bar accordingly since the alligator clips will register the touch because of their conductivity
* We will take advantage of the shake function and when the microcontroller senses that the board is experiences a shaking motion, we will have it print the space bar
* For the left and right click, we will attempt to use the photosensor as well as the temperature sensor to determine when the clicks are desired. Through experimentation we will determine what a good threshold would be and then apply it in the code. The left and right side will be assigned based on the criterion of ease of use (I.e. left click should be easier to use as it is used more often) and spatial location on the microcontroller (ideally would have left click on left side and right click on right side).
* We will need a lot of testing of each code in addition to testing how the different functions work together and influence each other, adjustments to the code will be made to improve compatibility between the various sensors after testing.

1. **Prototype**
   1. **Document the creation of your prototype: save well-commented code; take pictures or create diagrams of your hardware; document sources used (citations); etc**



#import libraries

from adafruit\_circuitplayground.express import cpx

from adafruit\_hid.mouse import Mouse

from adafruit\_hid.keyboard import Keyboard

from adafruit\_hid.keycode import Keycode

import array

import math

import time

import audiobusio

import board

#defining values for audio input

def mean(values):

return sum(values) / len(values)

def normalized\_rms(values):

minbuf = int(mean(values))

sum\_of\_samples = sum(float(sample - minbuf) \* (sample - minbuf) for sample in values)

return math.sqrt(sum\_of\_samples / len(values))

#setup audio input

mic = audiobusio.PDMIn(

board.MICROPHONE\_CLOCK,

board.MICROPHONE\_DATA,

sample\_rate=16000,

bit\_depth=16)

samples = array.array('H', [0] \* 160)

mic.record(samples, len(samples))

#setup tap detection

cpx.detect\_taps = 1

#setup keyboard for WASD control

kbd = Keyboard()

#setup mouse

m = Mouse()

cpx.adjust\_touch\_threshold(200)

while True:

#acquire value for magnitude of audio

mic.record(samples, len(samples))

magnitude = normalized\_rms(samples)

print(((magnitude),))

#define acceleration values

x, y, z = cpx.acceleration

#setting audio threshold to activate spacebar

if magnitude > 200:

kbd.send(Keycode.SPACEBAR)

#setting tap detection to activate left click

if cpx.tapped:

m.press(Mouse.LEFT\_BUTTON)

m.release(Mouse.LEFT\_BUTTON)

#setting light threshold to activate right click

if cpx.light < 3:

m.press(Mouse.RIGHT\_BUTTON)

m.release(Mouse.RIGHT\_BUTTON)

#setting accelerometer values to move mouse

if x > 4:

m.move(-5, 0, 0)

if x < -4:

m.move(5, 0, 0)

if y < 4:

m.move(0, -5, 0)

if y > -4:

m.move(0, 5, 0)

#setting touch pads to activate WASD keys

if cpx.touch\_A4:

kbd.send(Keycode.W)

if cpx.touch\_A3:

kbd.send(Keycode.A)

if cpx.touch\_A7:

kbd.send(Keycode.S)

if cpx.touch\_A1:

kbd.send(Keycode.D)

Referenced: *https://circuitpython.readthedocs.io/projects/circuitplayground/en/latest/api.html#adafruit\_circuitpla yground.express.Express.touch\_A1*

1. **Test and evaluate your prototype.**
   1. **Before you begin testing, write up the methodology you’re going to use. Your writeup should describe and explain how you are going to test your prototype and identify what would represent a successful test outcome versus an unsuccessful one.**

When the microcontroller works as a mouse cursor, has the functionality of controlling left and right mouse clicks, the space bar, and the WASD keyboard keys, we will have achieved success. If we can play a game which requires the use of those keys using the devices, it will constitute success. To test if our prototype and code are successful, the mouse cursor would need to move left when the device is tilted left, right when tilted right, up when tilted up, and down when tilted down; the left click would need to function when the device was tapped; the right click would need to function when the device was covered up eliminating light; the spacebar would need to function when the device picked up a large audio input; the WASD keys would need to function when each of the capacitive touch sensors were touched accordingly to how it was coded.

* 1. **Is your solution/design reasonable? Does it fully address the design goals? Does it do so minimally or does it do so well?**

Our solution was successful and reasonable. Using Python and the microcontroller we were able to develop a wearable assistive technology device that could control the computer cursor with both left and right mouse clicks as well as could have the WASD & Space keys function, all without the use of individual fingers.

* 1. **Are there aspects that are not functional and obviously need to be fixed? Are there aspects that may work but could possibly be improved?**

Some aspects that could be functioning better are the mouse movements and the right click. The mouse currently only moves at one speed and the click sometimes is triggered accidentally.

* 1. **Is it reliable? Can it be easily broken or caused to fail/crash? Are their special cases or conditions that cause it to not perform as desired?**

A special case that causes it to work incorrectly is when the device is turned upside down to move the cursor around. The accelerations that are being used get flipped around as well as the decreased level of light may trigger a mouse click. The environment being too loud could also trigger the click of the spacebar.

* 1. **Have you thoroughly checked your calculations and/or code for accuracy, consistent units, bug-free execution, etc.?**

Yes, we checked the code to make sure it compiled and there were no errors or bugs.

* 1. **Have you made assumptions in your design, calculations, or programming that are questionable?**

We assumed that the user of the device would have at least some functionality of their hands, enough to at least touch the alligator clips. We also assumed that the environment the device is used in, is going to be bright enough so that the function of the light sensor works properly. We also assumed that the environment is not going to be very noisy; quiet enough that it can pick up sound from the user to make the space bar function.

* 1. **Is there a possibility that a better solution might exist if enough time and effort were spent looking for one? (Hint: Yes!) How might you go about finding it?**

A better solution might exist. If we had more time, we might have been able to use other capabilities of the microcontroller to complete some of the requirements. We chose to use some of the abilities that we were comfortable with and that we knew we could successfully implement. Ideally, we would like to explore more of the controller's capabilities to see if we could design a device that worked flawlessly and precisely. Also, the more we familiarize ourselves with the Python language, the better the solution might become.

1. **Iterate to improve your prototype. Review what was learned in response to prompts 1–6 and repeat any and all prompts to achieve success and the best design considering the constraints within you must work.**
   1. **Document the iteration process. Explain what changes were made and why.**

As we worked along and tested the code for each function we needed, we continued to make small changes. We had originally planned to use the shake and temperature features of the controller, but as we worked through developing the prototype, we decided that the tap function and audio function would work more efficiently for what we were trying to achieve. As we wrote the code, we also made small changes to some of the threshold values. We needed to make the mouse move faster, increase the sensitivity of the mouse, and increase the sensitivity of the photo sensor so the device worked more effectively. We continued to make small changes as we developed the code and device design.

1. **Reflection**
   1. **Overall Assessment (0-100 and why)**

96, We think that our code functioned very well when tested on the in-class games and we used various sensors on the micro controller that made it into an intuitive mouse/keyboard that achieved what we set out to achieve.

* 1. **Technical Content**
     1. **What are you proudest of in terms of your technical work? What did you do particularly well?**

We used the tap sensor to simulate a left click and that worked surprisingly well and consistently. Additionally, our code is relatively succinct and achieves what it needs to in no more lines than necessary.

* + 1. **Where do you think there might be weaknesses or errors in your technical work? Are there still technical questions that you have? How would you find answers to those questions if you had more time?**

The velocity of the mouse was constant, and the accelerometer only changed the direction it moved in; this may be a weakness in our design. However, we found an optimal velocity for the mouse to move in which allowed for good control and not many errors. The technical questions we have would be how to integrate more precision in the mouse effectively. If we had more time, we could research the programming needed to make this work.

* 1. **Problem Solving and Teamwork**
     1. **What was the most serious problem that you had completing the assignment? How did you deal with it? How will you avoid or minimize similar problems the next time around? Should the instructor do more to help and, if so, what would you suggest?**

The most serious problem we faced was in trying to figure out why sometimes the alligator clips/capacitive touch pads were inaccurately registering that there was in an input and were causing a click of the WASD keys. We had to adjust the alligator clips and place tape over the touch pads so that there was no unintentional button triggering. Within the code, we also adjusted the threshold values to decrease the sensitivity so that the keys would only be triggered when we really wanted them to. Any similar problems we faced just required us to trouble shoot and problem solve with either the device or with the code. There were very few significant problems that we had to face. We thought Professor Henderson gave us the resources we needed in order to successfully complete the problem.

* + 1. **What, if anything, did you learn about problem solving by working through the assignment? If you did not learn anything, why do you think the experience failed to help you develop problem-solving skills and what would have been helpful?**

In regards to problem solving, we learned how to work as a team to overcome any obstacles we faced. For example, we had to our resources to find information and relevant code which made our device work as desired. There were times that we had to take a step back to refocus on what the problem was that we needed to solve, so that we could do it in the most efficient way. Also, not having much of a background using Python prior to the assignment, definitely forced us to work through the challenge.

* + 1. **What worked well for you in your groups? What issues or problems did you have?**

Overall, our group worked very well together. If we came across an error, we worked together to trouble shoot it. We also were very receptive of each other’s ideas and collaborated well to brainstorm ideas, design, and improve the device. The only issue we faced was making time to meet because our schedules are very busy.