**Lab 2**

**Basic Inputs and Outputs**

**Clayton Davidson - 000860643**

**February 2, 2022**

**P1**

Text

Description automatically generated

**P2**

**\*\*Please See p2.txt for Truth Table\*\***

**P3**

A screenshot of a computer

Description automatically generated with medium confidence

**YouTube Link -** [**https://youtu.be/ksXjLJ8\_x9A**](https://youtu.be/ksXjLJ8_x9A)

**P4**

Text

Description automatically generated

**\*\*Please See p4.txt for Truth Table\*\***

**P5**

Table, calendar

Description automatically generated

**Generated Boolean Expression:** (D!E!) + (DE) + (AD!E)

**Critical Reflection**

Being able to reverse engineer Boolean logic in the manner we did here, gives us the very beginnings of knowledge of how to understand and overcome logic gates in other devices such as integrated circuits in smart devices, home automation and industrial process systems. This can be used to make programs to run these systems(called firmware), or it can be used to create/analyze malicious/suspicious firmware.

This Lab helped me to better understand the relationship between physical(buttons/LEDs) and digital(I/O) signals. The evolutionary process from taking user input through the CLI, to physical input with Buttons, and further to sending output to a device and receiving feedback was very instructive and progressed in a way that was easy to follow. The ability to use this feedback in such a way to reverse engineer Boolean logic will make it easier in the future to understand complex internal logic, without having to know the original logic.

I had several minor problems with Python syntax, which were resolved throughout the process. I also had a problem in Part3 where, initially, my program would only re-assess the input values from the buttons when a button was pressed. My solution was to also create callback functions for button release, but we were shown a better solution in class, where you had the subprocess call inside a while loop to ensure than the state of button inputs was constantly reassessed. I also had some issues with Part4, creating a script wrapper to send/receive I/O when plugged into your BlackBox. This was resolved after some trial and error by including a sleep() function after the outputs were sent and before the input was received. This allowed the BlackBox adequate time to process my given inputs and send a valid output back. Without this, the delay in processing/communication speed would give me a false positive on all output states. One other minor problem in Part4 was sending object values instead of the desired integer value of that object, which was easily resolved by adding ‘ .value ‘ to all the object references.