**Milestone Two: Serial Sensations**

**CS-350**

Damean Murphy-Short

Southern New Hampshire University

January 20, 2025

The following answers are provided in response to lab questions from the Milestone Two GPIO UART Lab Guide

**1. Why is the string.encode() method used when using the write() method for the serial connection?**

The write() method expects input to be of type bytes, bytearray, or memoryview and the encode() method of the string data type converts a string into the bytearray data type. Another practical reason for this is that there are several ways to encode text, with different possible bit-lenghts for a single character, so using the same encoding in both the server and client ensures that the original message is preserved.

**2. Why is the decode() method used when data is read()?**

Data read from the serial port is read as a binary stream of bytes. To understand these bytes as characters, it is necessary to decode them using an encoding such as UTF-8 or ASCII. As in the answer above, we wish to ensure that the server and client are using the same encoding.

**3. Why is there a try/except block present in the modules reading and writing serial data?**

The try/except block catches one exception: KeyboardInterrupt. This allows the user to end the program flow manually by entering a ^C character. Since the exceptions that may possibly be raised by the write() and read() methods are not handled, the program will fault upon reaching them, which is acceptable for our use-case, though it may be advisable to handle these exceptions to ensure that all GPIO cleanup is executed before program exit, as will soon be discussed.

**4. Why is it necessary to return all GPIO pins to their original state before program exit?**

When a GPIO output port is set high, a voltage is applied to the pin. It is possible the person operating the system could not be aware of the state of the pin before the program is closed if no cleanup is processed, leaving a voltage that is not observable to the human eye active in a high state. This has the potential effect of damaging components that are connected to this pin that do not expect to receive any voltage or of damaging the pin itself if it comes into contact with a ground connection or another output voltage. Therefore, if the purpose of the program is not to supply voltage to some device and then cease operating, best practise is to ensure all pins are set to 0V and as inputs before exiting.