

MATLAB Arduino Library for Seven Segment Display

This library was developed for an independent study course. The goal was to develop a MATLAB function that could address a seven segment display via an Arduino Uno. The chosen display has four characters and functions through multiplexing. This means that all of the characters are addressed at once and the one to enable is chosen by closing a corresponding ground. By rapidly changing the character enabled, the display is multiplexed and the entire display appears as continuously active. This method reduces the number of pins required to create such a display. Normally this type of display, driven by an Arduino, is done through the use of a backpack chip that automatically indexes the characters. The intent was to discover whether this could be done directly via the Arduino.

The functions written allow the numbers zero through nine to be displayed, as well as a decimal, for each character. The main function takes a series of arrays as inputs that include the Arduino pins connected to each pin of the display and the characters to be displayed. This setup uses all available digital pins on the Arduino Uno. The analog pins may have also been utilized but this presents issues in assigning and addressing them using the functions. If the user mixes analog and digital pins the function must either also require the pins used to be specified or sort them itself. This adds complexity to the function and makes it more difficult to use.

The function multiplexes through each character on the display correctly, however, it fails to do this fast enough to create the illusion of a continuous display. Instead, it cycles through each character at a perceptibly slow speed. The MATLAB Arduino add-on works by installing software on the Arduino that allows commands to be sent to it via the USB cable. An attempt was made to increase the multiplexing speed by increasing the baud rate of the arduino. It was found that the Arduino object created already defaults to the maximum. However, the speed of the multiplexing could be reduced by decreasing the baud rate, demonstrating that it does have an effect on the display. The USB port on the Arduino uses the 2.0 specification with a max speed of 480 Megabits per second. This, being substantially higher than the frequency of the Arduino, was not the limiting factor. It was observed that selecting a baud rate of 9600 instead of 115200 did slow the multiplexing speed but not by a factor proportional to the change. Instead this factor of 1:12 only reduced the speed by about half. This indicates that the limiting factor must lie elsewhere in the system.

Another culprit may simply be the way MATLAB and Arduino communicate. Since the code is not run natively on the Arduino, the time it takes to send a pin write command to the Arduino may be significant. Multiple forum posts complaining of high latency when addressing the Arduino corroborate this. Investigating this further would require greater understanding of the way MATLAB and Arduinos work and analyzing the libraries used by the add-on. For now, the conclusion is that the add-on is likely the limiting factor. It may be possible to get the display to function as intended with natively run code on the Arduino. Although the Arduino Uno has sufficient pins to address the display, the MATLAB Arduino add-on is not the ideal way to run it.