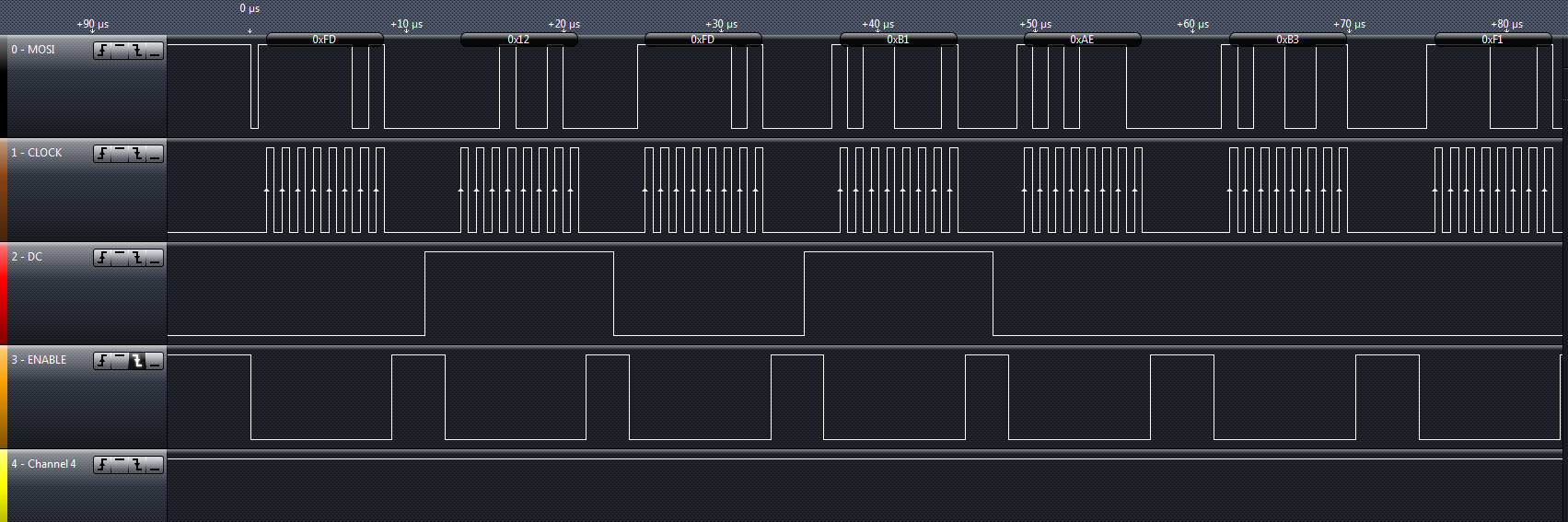
**Waveform**

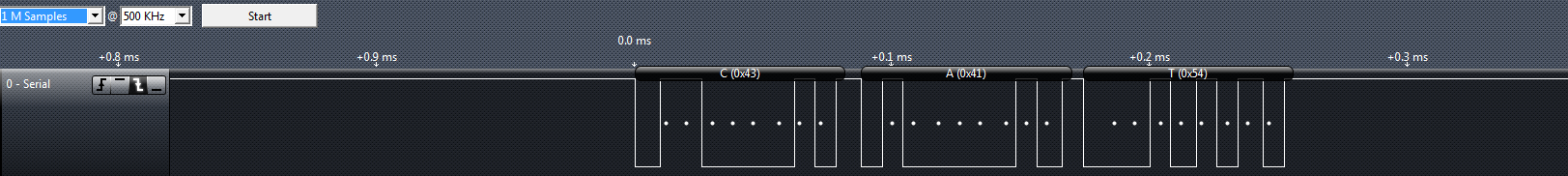
**SPI**



*Explanation:*

WriteCommand sets the DC to low and outputs a FD (which is the COMMANDLOCK), after writeData writes 0x12 on high (When DC is high).

**UART**



Stanley Chen

Kevin Fu

EEC 172

**Lab Writeup**

In this lab, we are implementing further improvements upon our first lab. We will be building a transmission of between two OLEDs. It will simulate a very basic text-messaging program.

During this lab, our initial difficulty was translating the code from C++ to C. It was hard to understand which ones were valid in C++ or in C. Since C++ is object-oriented and C is not, we had to remove all the classes and any abstract data types. It was mostly a lot of trial and error between removing certain parts of code and dealing with compilation errors. After a lot of tweaking, we finally got the OLED to turn on and complete part 1. For part 2, there was not too much of a problem since it is essentially our first lab. The only part difficult about this part was implementing a code that kept printing things out consecutively and creating the pause function. There probably were many ways to implement the pause function, but we chose to use polling for our program. The last part was a little more difficult in that it required us to understand how the interrupts happened for the UART. I was trying to use both UART0 and UART1 from the example together, but we only needed one (we chose to use UART1).The transmission was first hard to understand, but once we understood it, it was relatively basic and only required a small addition to our code (~30 lines or less).

Overall, this lab assignment gave us an actual useful implementation of the first lab that we did. It showed