When SW3 is detected as being pressed and released (lines 56-62), the MSP430 generates 4 packets of data that are sent to the Nokia 1202 display, causing a vertical bar to be drawn. Complete the following table by finding the 4 calls to writeNokiaByte that generate these packets. In addition, scan the nearby code to determine the parameters being passed into this subroutine. Finally, write a brief description of what is trying to be accomplished by each call to writeNokiaByte.

| **Line** | **R12** | **R13** | **Purpose** |
| --- | --- | --- | --- |
| 66 | #NOKIA\_DATA (0x01) | 0xE7 | Draw the split beam on the buffer. |
| 276 | #NOKIA\_CMD (0x00) | Upper byte: 0xB0 Lower byte: Row number | Converts a row number to the proper format |
| 288 | #NOKIA\_CMD (0x00) | Upper byte: 0x10 Lower Byte: Upper byte of given column number | Converts the column number to the proper format. |
| 294 | #NOKIA\_CMD (0x00) | Upper byte: 0x00 Lower Byte: Unchanged from above | Copy Buffer to screen |

Configure the logic analyzer to capture the waveform generated when the SW3 button is pressed and released. Decode the data bits of each 9-bit waveform by separating out the MSB, which indicates command or data. Explain how the packet contents correspond to what was drawn on the display. Be specific with the relationship between the data values and what and where the pixels are drawn

| **Line** | **Command/Data** | **8-bit packet** |
| --- | --- | --- |
| 66 | Data | 11100111 (split beam) |
| 276 | Command | 10110001 |
| 288 | Command | 00010000 |
| 294 | Command | 00000001 |

Next, setup the Logic Analyzer to capture the RESET signal on a falling edge. Measure the duration that the RESET line is held low in the initNokia subroutine. Hint, the code to hold the reset line low can be found on lines 93-100. How many counts does the firmware loop count down from? Using the delay you just measured and the number of counts, calculate the amount of time each iteration of the delay loop consumes.

The duration that the reset signal is active is 6.6875 μs. Since the counter counts from #0FFFFh, the amount of time per loop is: