Below are the waveforms for the wakeup sequence and going through the auto addressing process. We have one base device, with one stack device.



This frame is when I send along the first read command. I can see the data coming back from the Stack device to the bridge device, but it does not come through over the MISO line. SPI\_READY never triggers back high, it stays low until it times out.



Here is the code I am using the send the first read command. We are using an STM32, and using SPI HAL Libraries to send and receive messages. I am trying to follow this format





This is how I have my read messages set up, just as arrays that I send out over SPI. For the read command, I send the receive, which is the first read command, padded with zeros until its 128 bytes long, so that I am transferring 128 bytes to the device. Then I check if SPI\_READY, but SPI\_READY never triggers back low again



I was trying to send the read command using the same format as all my write commands, shown below, but I ran into the same issue, which is what prompted me to split up the send / receive above. I also was told that if I used a transmit/receive command I would need to zero pad out the message so that it has the clock to send data back out on.



I tried a different approach as well, where I did SPI transmit, then SPI receive separately from each other. We ran into the same issue, where SPI\_READY never triggered high again after triggering low



