3-3 Journal: Peripheral Interfaces in Embedded Systems

Joel Meza

Professor Ricky Sethi

January 21, 2022

CS 350 – Emerging Systems Architectures & Technologies

Southern New Hampshire University

I have chosen three peripheral interfaces within embedded systems that are frequently used which are known to be “UART, SPI, and I2C” communication protocols. I’ll start saying which one is the slowest to fastest peripheral interfaces when considering them in embedded system and the slowest but still frequently used from all three is “UART”. Compared to “I2C” and “SPI”, UART communication protocol has two UART devices to transmit and receive data where it’s not a clock timer when operating it but does need baud rates from both UART as mentioned to be at minimum of ten percent each to avoid any data loss. Now, between both “I2C” and “SPI”, the I2C communication protocol is a bit slower than the SPI due to being half-duplex because it can’t separate lines to transmit and receive data. Although the I2C is faster than UART, because of ACK/NACK bit to discern or figure any changes in data during transmission.

However, I do believe that SPI communication is faster compared to the I2C and UART because of its full-duplex transmission which separates lines quicker and more discerning to transmit and receive data from an embedded system. The differences according to the number of wires each peripheral interface got is the duplexes and how many master and slaves they all got. So, UART communication protocol contains one wire, I2C communication protocol contains two wires, and SPI communication protocol contains about four wires. Although, the UART is common to have one master and one slave connectivity, I2C has several masters’ and slaves’ connectivity, and SPI got one master and several slaves’ connectivity.

In conclusion, comparing the advantages and disadvantages from these peripheral interfaces is by means of the transmission and what they are used commonly for in an embedded system. If an embedded software engineer or team would need to connect sensors to a microcontroller but have more than 127 actuators, then an “SPI” communication protocol will be smartest choice from all of them. On the other hand, “a UART is often commonly used as a form of device-to-device communication from a computer and microcontroller applications” (Hopkins, 2021). Therefore, if an embedded software engineer or team need to connect sensors to a microcontroller but have less than 127 actuators and not considering a complex transmission connectivity, then the best choice of all three peripheral interfaces to be used is I2C communication protocol. Depending on the circumstances of the project, it will come down to what is being communicated and what can be the most efficient and better way to work on an embedded system.

**Citations**

Hopkins, J. (2021). *I2C vs SPI vs UART – Introduction and Comparison of their Similarities and Differences.* Retrieved From:[**https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-comparison-similarities**](https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-comparison-similarities) **-differences/#:~:text=Also%2C%20both%20UART%20and%20SPI,in%20computer%20and%20microcontroller%20applications.**