**Journal: Peripheral Interfaces in Embedded Systems**

Justin Starr

Department of STEM

CS 350 – Emerging Systems Architecture & Technologies

Professor Mike Prasad

January 28, 2024

**Journal: Peripheral Interfaces in Embedded Systems**

Identify three interfaces to compare and contrast and explain the differences between the three interfaces chosen. Then explain why you would use one interface over another.

**SPI – Serial Peripheral Interface**, according to Total Phase, Inc. (2024), is a serial communication protocol used in embedded systems for high-speed data exchanges between devices on a bus. Including at least four signals, the SPI protocol operates using a master-slave paradigm. The four signals include a clock (SCLK), a master output/slave input (MOSI), a master input/slave output (MISO), and a slave select signal (SS Signal). The four signals are shared by all devices that are on the bus. The master device generates the clock signal (SCLK) for synchronization between devices on the bus, and the MOSI and MISO signals are used for data exchange. Each slave device that is added to the bus has its own slave select line. The master-slave pulls on a slave’s slave select line to select a device to communicate with. SPI supports full-duplex communication which means that the master and the slave can transmit data at the same time.

The reason why a developer would choose to use SPI is because it is a good choice for high-speed data transfer applications, such as Digital-to-Analog Converters (DACs), displays, and flash memory (Prodigy Technovations, Pvt Ltd., 2024).

**I2C – Inter-Integrated Circuit**, according to Total Phase, Inc. (2024), is often used in embedded systems, I2C is a simple communication protocol used as a way to exchange data between one or more master slaves and one or more slave devices. It is a bidirectional two-wire serial bus, one wire for serial clock (SCL), and one wire for serial data (SDA) that is used to manage data between devices connected to the bus, bit by bit. I2C is considered synchronous, which means it operates using a serial clock that is driven by the master device. The typical data transfer rate of the I2C protocol is 100 kbps, however, when configured in “fast mode” or “ultra-fast mode”, data speeds of up to 5 Mbps are possible.

The reason why I2Cs would be a good choice is because they are ideal when connecting multiple devices to a single controller, such as Real-Time Clocks (RTCs), sensor arrays, and EEPROMs (electrically erasable programmable read-only memory), which is a type of non-volatile memory (Prodigy Technovations, Pvt Ltd., 2024).

**UART – Universal Asynchronous Receiver/Transmitter**, according to Total Phase, Inc., (2024), “is a physical circuit in a microcontroller or single integrated circuit (IC) that is used to implement serial communication between devices in an embedded system.” Its primary purpose is to transmit and receive data and in UART communication, two UARTs directly communicate with each other, and only two wires are required for this communication to occur. Data flows from the transmit (Tx) pin of the transmitting UART to the receive (Rx) pin of the receiving UART. Including half-duplex and full-duplex operations, UART supports bidirectional data transmission, and it is also asynchronous.

For many applications, UART is a good choice, especially for those emphasizing an easy implementation, and for robust, real-time communication (NEXTPCB, 2024). It is a cost-effective choice and has been universally accepted as a valuable tool for engineers and developers. UART is a good choice when communication between two devices needs to occur over great distances. No clock is needed with UART and a parity bit is used to allow for error-checking.

**References**

NEXTPCB. (2024). Understanding UART Communication.

https://www.nextpcb.com/blog/understanding-uart-communication

Prodigy Technovations Pvt Ltd. (2024). I2C vs SPI: Difference Between I2C and SPI.

[https://prodigytechno.com/i2c-vs-spi/#:~:text=%E2%97%8F%20I2C%20is%20half%20duplex%20communication%20and%20SPI,clock%20stretching.%20%E2%97%8F%20I2C%20is%20slower%20than%20SPI.](https://prodigytechno.com/i2c-vs-spi/" \l ":~:text=%E2%97%8F%20I2C%20is%20half%20duplex%20communication%20and%20SPI,clock%20stretching.%20%E2%97%8F%20I2C%20is%20slower%20than%20SPI.)

Total Phase, Inc. (2024). I2C vs SPI vs UART - Introduction and Comparison of their

Similarities and Differences. <https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-comparison-similarities-differences/>