I2C vs. I3C: What Are the Differences?

Published: 2022-05-04 **Updated:** 2024-01-08
Brendan Murphy

Introduction of I2C

I2C, or "Inter-Integrated Circuit," has become a de facto standard in the embedded systems industry since its inception in the 1980s by Philips. As the rapid adoption of electronic devices became more mainstream in the late 1900s and early 2000s, a need arose to make devices smaller and more portable. After all, people did not want to carry around a cellphone the size of a masonry brick everywhere they went. People wanted smaller devices that they could take with them on the go. With the new challenge of shrinking devices came the need to cram more systems into a smaller space. How do you add more with less space? I2C took a shot at answering that question.

I2C Overview

The I2C protocol simplified connected devices on a board by limiting the number of wires needed to connect embedded systems on a printed circuit board to just two lines; an SDA and SCL line, also known as a serial data line and serial clock line, respectively. The SDA line is used to send data across the bus while the SCL line synchronizes data transfer between master and slave devices. Simplifying a system to only two wires was a massive win for board design. It allowed for simple, clean, and compact board layouts resulting in more systems with less space, precisely the objective when making devices smaller and more compact.

I3C Inception

As technology improves over time, adaptations and enhancements to serial communication protocols often take place. For I2C specifically, the I3C protocol was designed as its successor to address some of its limitations. I3C, or "Improved Inter-Integrated Circuit," was introduced to the market by the MIPI Alliance in 2016, 35 years after the introduction of

I2C. The protocol was introduced to the market to improve upon the already existing I2C protocol. Some obvious improvements include greater speeds and more energy-efficient operations; less power is needed to do more. I3C maintained the two-wire bus characteristics of I2C but expanded upon other areas where the protocol was lacking.

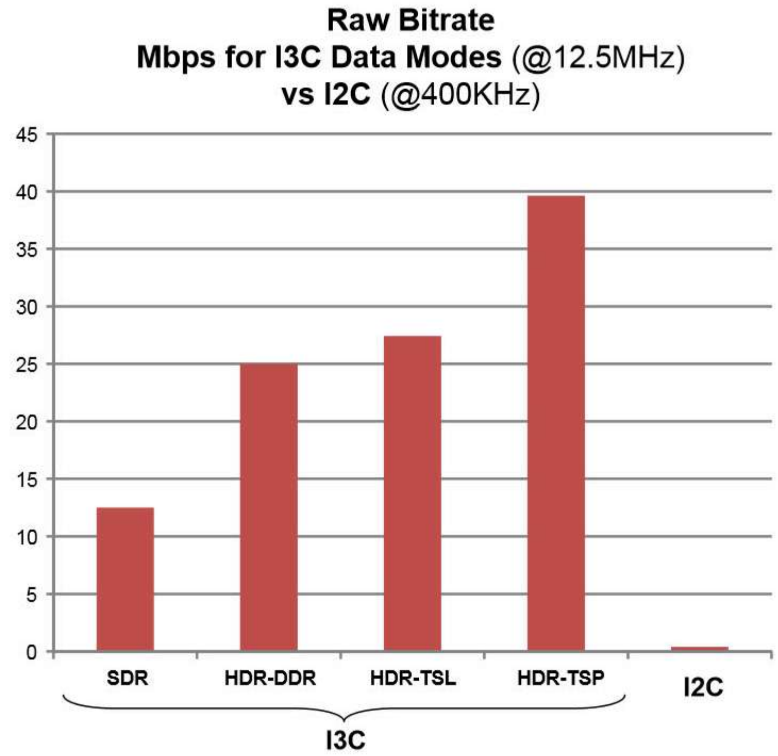
Faster Speeds

One of the improvements I3C has over I2C is its ability to send data at fast clock speeds. Historically, I2C did not have to be an extremely fast protocol as it was generally used in sensors that don't often require much data throughput to operate efficiently. However, as sensors and embedded systems have become more complex and sophisticated, a need for faster data throughput arises.

I2C operates in five main modes:

- Standard Mode: 100 kHz
- Fast Mode: 400 kHz
- Fast Mode Plus: 1 MHz
- High Speed: 3.4 MHz
- Ultra-Fast Mode: 5 MHz

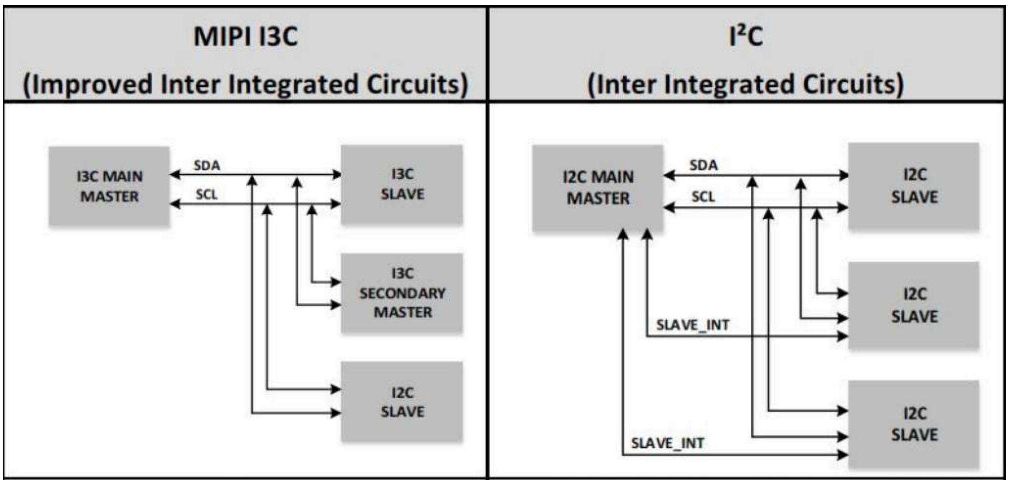
I3C, on the other hand, has a Standard Data Rate of 12.5 MHz, with new versions of I3C supporting data rates up to 100 Mbps. Needless to say, I3C supports speeds that far surpasses that of I2C.



I2C vs I3C Raw Bitrate Comparison. Source: MIPI Alliance

Backward Compatibility

When developing I3C, the ability for the protocol to be fully backward compatible was a significant focus. I3C applications can operate with I2C slave devices as well as native I3C devices. This backward compatibility allows engineers time to phase out their existing I2C applications. This phase-out time is necessary for embedded systems engineers because it makes the adoption of the new protocol much easier and streamlined.



I3C vs I2C bus circuits. Source: MIPI Alliance

Dynamic Addressing

I2C has a device addressing technique called static addressing. This, in simple terms, means that manufacturers lock in their device addresses in production. One of the drawbacks of this is that vendors may use the same address as another vendor for a product that uses I2C, causing issues during integration. During bus initialization, the I3C controller assigns a 7-bit dynamic address to each device on the I3C bus, thereby eliminating the problems associated with duplicate addresses because each I3C device receives a unique address from the master.

Hot-Join

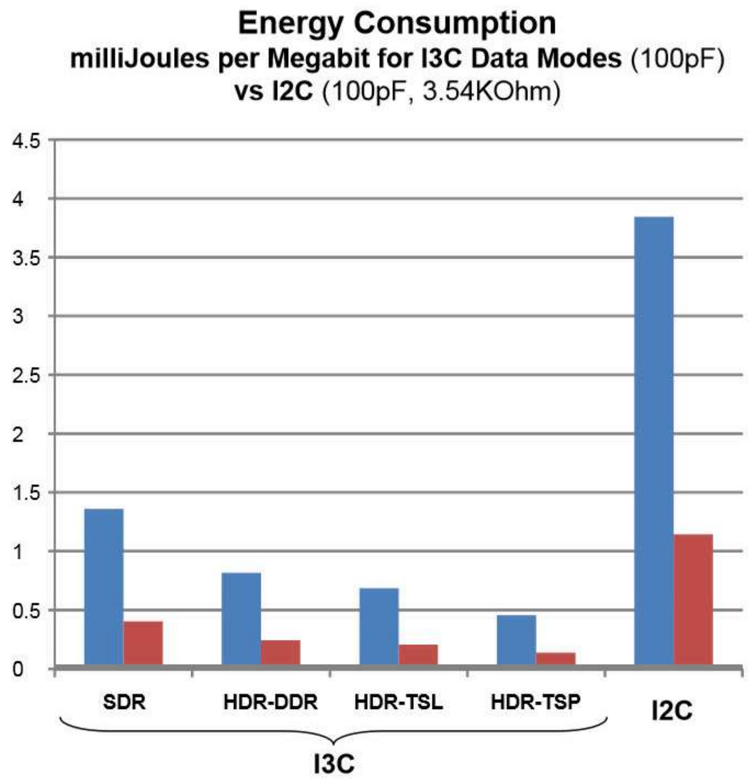
Another new feature to I3C is the protocol's ability to allow other slave devices to be added or taken away from the system with no interrupts to the system as a whole. The system is called hot-join. This mechanism allows slaves to join the bus after the bus is already configured.

In-band Interrupt

The I3C interface uses a push-pull clock line and an open-drain data line for operations. The data line allows slaves to take control and initiate interrupts when needed. This feature is unique to I3C as this same application on the I2C bus would require a third line. I3C slaves can request an interrupt when the bus is idle, whereas I2C slaves cannot. I3C assesses conflicts between multiple slaves by allowing the lowest assigned address to win.

Power Efficiency

Another attribute important to any new technology is its ability to manage power consumption better than previous product iterations. Computers need to be faster but use much less energy than the prior model; new iterations of bus protocols are no different.



I2C vs I3V Energy Consumption. Source: MIPI Alliance

Due to a change in output method from an open drain method to a push-pull output, I3C can be much more energy efficient than I2C. An open-drain process requires pull-up resistors that, when activated, require a significant amount of power to operate. Push-pull operations do not require pull-up resistors to work, meaning the method can reduce energy consumption because it does not need to power any external resistors to function.

Another factor that plays into the power-saving quality of I3C over I2C is its data transfer speed improvements. The benefits here are simple to understand. Because messages can now be sent over the I3C bus at significantly faster rates, the bus can turn on and shut off target devices much quicker than was possible with I2C. This process saves a noticeable amount of power consumption compared to I2C.

Total Phase I2C Debugging

Total Phase offers both host adapters and protocol analyzers to support the debugging and development of the I2C protocol. The host adapters allow users to interface directly with an I2C system and program I2C data. Whereas protocol analyzers non-intrusively monitor traffic that is being sent over the bus in real time.

Total Phase Host Adapters

Total Phase offers two host adapters for I2C: the Promira Serial Platform and Aardvark I2C/SPI Host Adapter. Both of these tools interface with the I2C protocol with the main differentiators being in terms of speed and expandability.

The Aardvark I2C/SPI Host Adapter is able to act as an I2C master and slave up to 800 kHz. If higher speeds are required, the Promira platform becomes more suitable with its ability to signal up to 3.4 MHz as a master and slave. With this type of speed, the Promira platform is better suited for interfacing with high-speed I2C devices.

Aardvark I2C/SPI Host Adapter

The Promira platform also has an integrated level shifter built into the hardware allowing it to change voltage levels via software control, while the Aardvark adapter is limited to 3.3V (5V tolerant) without external accessories. The Promira platform offers up to 12 GPIOs whereas the Aardvark adapter can enable up to 6 GPIOs.

Promira Serial Platform

Total Phase Protocol Analyzers

For bus monitoring, Total Phase offers protocol analyzers that provide in-depth information about a specific bus. For I2C, the Beagle I2C/SPI Protocol Analyzer is the perfect fit to analyze I2C data. Simply attach the

Beagle I2C/SPI analyzer to an I2C bus and instantly watch as the tool streams the traffic being sent over the bus in real time in the Data Center Software.

Beagle I2C/SPI Protocol Analyzer

Data Center Software

Being able to analyze the I2C bus in true, real time provides time-saving advantages to the engineer. Unlike many other bus sniffers that require long download times, the Beagle I2C/SPI analyzer gives the user the information as the transaction is taking place on the bus. This gives priceless insights into how the system is working, giving engineers greater knowledge on how to debug their systems.

Conclusion

Although very similar to I2C in many ways, I3C builds upon the widely-used I2C protocol to modernize it and make it better suited for future applications. With its speed improvements, power consumption reduction, and other quality enhancements, I3C is quickly becoming a standardized protocol in upcoming applications.