MEMS and Sensors Whitepaper Series

Introduction to the MIPI I3C Standardized Sensor Interface

August 2016

Whitepaper Topics: sensors, digital communication (MIPI I3C, I²C, SPI), standardized sensor interface, unified sensor link, high-speed data rate, sensor data batching, sensor hubs, mobile, wearables and IoT

About Us: MIPI Alliance (MIPI) develops interface specifications for mobile and mobile-influenced industries. There is at least one MIPI specification in every smartphone manufactured today. Founded in 2003, the organization has more than 270 member companies worldwide and 14 active working groups delivering specifications within the mobile ecosystem. Members of the organization include handset manufacturers, device OEMs, software providers, semiconductor companies, application processor developers, IP tool providers, test and test equipment companies, as well as camera, tablet and laptop manufacturers. For more information, please visit www.mipi.org.

Contact Information:
Melanie Cole (melanie.cole@mipi.org)
Manager, Integrated Marketing & Membership
MIPI Alliance (www.mipi.org)

Introduction

The total global sensor market is expected to reach $154.4B by 2020[1]. $10.46B of this will be made up by integrated sensors that include decision making, logic functions and digital communication. Sensors are experiencing an unprecedented growth, from $650M in 2012 and with an expected compound annual growth rate of 36.25% through 2020[2]. A major driver has been the adoption of low-cost, small form factor sensors in smartphones, tablets and a growing number of wearables applications. The original iPhone was introduced in 2007 with advanced sensing capabilities (i.e. ambient light, accelerometer and proximity) that provided an innovative and improved user interface. Since then, the adoption of sensors has rapidly escalated and the largest sensor manufacturers now ship billions of units per year for mobile and consumer applications. This trend is continuing. Smartphone providers are trying to gain a competitive advantage by adopting new sensors and sensing technologies for improved user experiences. Increased sensor content allows more complex features through sensor fusion, where sensor data from multiple sensors is aggregated and analyzed to provide functions that cannot be achieved with a single sensor. Advanced functionality such as dead-reckoning, augmented/virtual reality and others require better performing, faster and sometimes duplicate sets of sensors. New sensing technology is also being developed. A possible next frontier could be the addition of gas and chemical sensors to monitor air quality and toxicity levels in the world around us.

The proliferation of sensors in mobile devices requires application processors and/or sensor hubs with an increased number of logic pins used for sensor communication and control. In a typical application, multiple digital communication interfaces are used along with supporting logic lines for dedicated interrupt and sleep signals. Top tier smartphones include 10 or more sensors and a critical point has been reached where 20 or more logic signals are required. There are other complicating factors as well. The de facto communication standard for sensors in mobile and consumer applications is I²C. I²C requires only two signal lines (clock and data), but has several shortcomings, including the inability for sensor slaves to initiate communication, an overhead protocol that reduces throughput and pull-up resistors that limit clock speed and increases power dissipation. Another commonly used standard is the serial peripheral interface or SPI. SPI requires four communication lines and is used where large amounts of data needs to be transferred, such as clearing data batches from first in, first out (FIFO) buffers. To its disadvantage, SPI lacks a clearly defined standard that has resulted in many different implementations.

There is no consistent method for interfacing to sensors, which causes considerable integration challenges. Device and platform designers are faced with digital interface fragmentation and have to deal with I²C, SPI, UART and others. What if there was a single, scalable, cost-effective and clearly defined standard communication interface for sensors? And what if this standard could unite the best of I²C and SPI while adding new functionality that the current standards are lacking? In 2013, MIPI Alliance formed a Sensor Working Group with a goal to develop an I²C compatible interface with sensor-focused, differentiated features. An extensive industry survey was performed in conjunction with the MEMS & Sensors Industry Group (MSIG) to collect information about what a new standard should include. The culmination of this survey along with the ongoing efforts of the Sensor Working Group is the brand new MIPI® Alliance Specification for I3C℠ Improved Inter Integrated Circuit (MIPI I3C) sensor interface standard.

[1] "Global Market for Sensors to Reach Nearly $154.4 Billion in 2020; Image Sensors Moving at 11.7%", BCC Research, July 2014
[2] "Smart / Intelligent Sensor Market by Type, Application & by Geography, 2013 – 2020", MarketsAndMarkets, March 2014

MIPI I3C Scope and Purpose

The MIPI I3C interface is an evolutionary standard that improves upon the features of I²C, while maintaining backward compatibility. This standard offers a flexible multi-drop interface between the host processor and peripheral sensors to support the growing usage of sensors in embedded systems. The main purpose of MIPI I3C is threefold: 1) to standardize sensor communication, 2) reduce the number of physical pins used in sensor system integration and 3) support low-power, high-speed and other critical features that are currently covered by I²C and SPI. Developing the MIPI I3C interface has been a communal effort to identify the real market needs, define a standard and merge technology contributions from leading SoC, sensor hub and sensor vendors. During the initial phases of the MIPI I3C development, multiple proposals were considered and technically graded to come up with a standard that satisfies a broad range of applications that extends beyond the smartphone. The subsequent development work has been a broad effort from many MIPI Alliance members[3]. The MIPI I3C standard is currently being finalized and will be released later in 2016, exclusively to MIPI Alliance members.

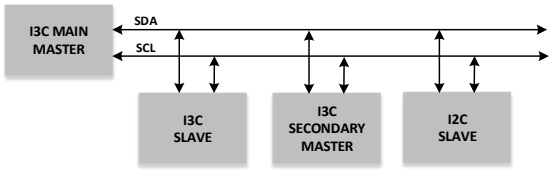
MIPI I3C Fact Sheet	
MIPI I3C Standardized Sensor Interface	
 Two-wire communication interface, clock (SCL) and data (SDA)	
Number of gates	< 2,000
Bandwidth	> 33 Mbps
Features	In-band interrupts In-band command codes Dynamic addressing Multi-master / multi-drop Hot-join support Backward compatible with I²C

Table 1: *MIPI I3C standardized sensor interface at a glance.*

MIPI I3C was initially intended for mobile applications as a single interface that can be used for any sensor. Modern smartphones that include a multitude of sensors and a slew of supporting logic lines are pushing the boundaries of both I²C and SPI. MIPI I3C will accommodate many sensors on the same communication bus, while eliminating additional logic signals needed to support interrupt or sleep mode functionality. The MIPI I3C standard is useful for other applications than smartphones. It offers high speed data transfer at very low power levels, which is highly desirable for any embedded system. Wearables is a great example where multiple sensors are used in a very limited physical space and with stringent power restrictions. Over time, the MIPI I3C could conceivably become much more than a standardized sensor interface and develop into a de facto bus communication standard for touch sensing, always-on and low resolution cameras, acoustics, environmental sensors and transducers that currently use I²C, SPI, UART and others.

[3] AMD, Broadcom, Cadence, Intel, InvenSense, Knowles, Lattice Semiconductor, MediaTek, Mentor Graphics, Nvidia, NXP, Qualcomm, QuickLogic, Sony, STMicroelectronics, Synopsys, VLSI Plus, ZMDI (now IDT) and others.

MIPI I3C Fundamentals

The MIPI I3C interface uses an I²C-like interface with an open drain data line (SDA) and a push-pull clock line (SCL). The open drain (i.e. open collector) SDA line allows for slaves to take control of the data bus and initiate interrupts. The push-pull SCL line is used by the master to clock the communication bus up to 12.5 MHz. MIPI I3C supports multiple classes of devices including main master, secondary master, MIPI I3C slave and I²C slave. The master can dynamically assign 7-bit addresses to all MIPI I3C devices while supporting the static addresses of legacy I²C devices. This ensures full compatibility between MIPI I3C and I²C. The MIPI I3C interface represents a shift in power performance while providing greater than an order of magnitude improvement in speed over I²C. I3C offers four data transfer modes that, on maximum base clock of 12.5MHz, provide a raw bitrate of 12.5 Mbps in the baseline SDR default mode, and 25, 27.5 and 39.5 Mbps, respectively in the HDR modes. After excluding transaction control bytes, the effective data bitrates achieved in each mode are 11.1, 20, 23.5 and 33.3 Mbps, respectively, protected by I3C's basic error detection mechanisms. The bar charts in Figure 1 compare the energy consumption (per bit) of the various MIPI I3C modes with I²C (left) and the corresponding raw bitrates (right). Based on these results, the MIPI I3C is a more power efficient interface even in the I²C-compatible mode. The MIPI I3C ternary HDR-TSP mode is the fastest and most power efficient mode, supporting effective data bitrates over 33 Mbps.

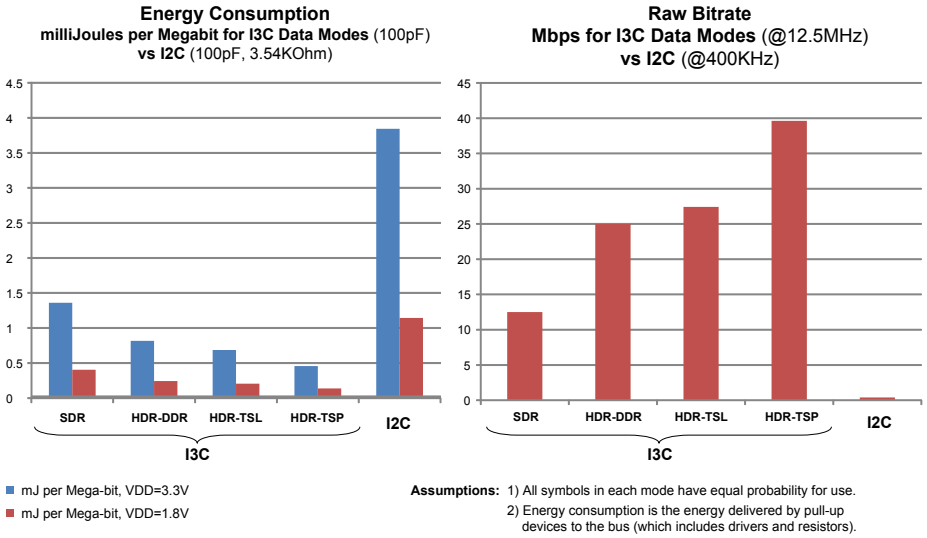


Figure 1: *Energy consumption for MIPI I3C modes in comparison to I²C (left) and raw bitrates for MIPI I3C modes compared to I²C (right). Image from MIPI Alliance.*

A very useful feature allows MIPI I3C slaves to initiate in-band interrupts, which currently requires a dedicated signal line for both I²C and SPI devices. The in-band interrupt feature enables slaves to issue a "start" when the bus is available (i.e. idle). The master provides an interface clock for the slave to drive its master-assigned address onto the bus to initiate an interrupt. If there is a conflict where multiple slaves are trying to initiate an interrupt simultaneously, the lowest assigned address wins by arbitration. The master can acknowledge (ACK) the interrupt and restart the bus or continue to clock out data from the slave. A not-acknowledge (NACK) can be sent to end the communication. For example, an ambient light sensor is commonly used to control the backlight brightness of a display in mobile devices. If the light conditions change, the sensor will send an interrupt to the system and requests that sensor data is

sent to the master. When light conditions are stable, there is no need to inquire the sensor and use power for unnecessary sensor data transmissions.

MIPI I3C slaves are allowed to join the bus after it has already been configured. This is called hot-join. Sensors that are connected on the same bus can be powered off until they are needed. Hot-joining should adhere to specific electrical limits and not disturb the MIPI I3C lines. A set of common command codes (CCCs) has been defined for standard operations like enabling and disabling events, managing MIPI I3C specific features (dynamic addressing, timing control etc.) and others. These CCCs can be broadcasted (sent to all devices) or directed at a specific device on the bus. Power efficiency and performance are crucial in sensor applications. Always-on sensors and sensor hubs are constantly accumulating sensor data even while the main application processor is idle (i.e. low-power mode or deep sleep). Accumulated sensor data is commonly organized in batches that need to be periodically and quickly transmitted between sensors, sensor hubs and application processor to minimize power consumption. The industry has favored SPI for high-speed transmission of batched sensor data, but SPI is more complex and has higher pin count than I²C.

MIPI I3C Sensor Interface Standardization

I²C and SPI have become synonymous with digital communication for sensors. I²C or inter-integrated circuit was originally developed by Philips Semiconductor (now NXP Semiconductors) back in 1982 as a simple communication bus for building control electronics. The initial version of I²C ran at a 100 kHz bitrate and faster bitrate versions were adder later. To this day, the 400 kHz version is the most widely used and the primary communication protocol for sensors. I²C uses two bidirectional open-drain signal lines, one data line (SDA) and one clock line (SCL). The I²C bus is designed for one or multiple masters and one or more slaves (see Figure 2). Slave devices can be individually addressed by a 7-bit (most common) or 10-bit addressing scheme. The master generates the clock and initiates the communication with the slave(s). However, an I²C slave cannot initiate communication and this is a major shortcoming that the MIPI I3C standard has resolved with in-band interrupts.

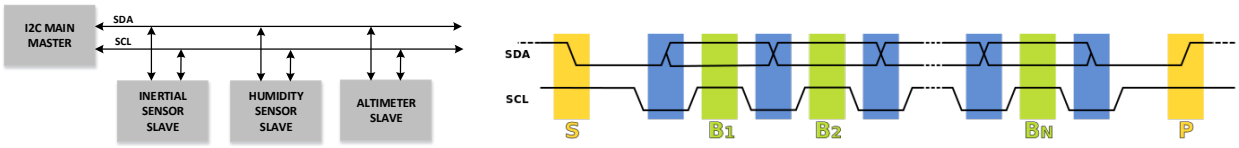
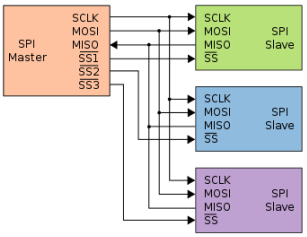


Figure 2: *I²C sample schematic with one master and three slaves, inertial sensor, humidity sensor, pressure sensor and a microcontroller (µC) (left) and sample timing diagram (right). S = Start, B1 to Bn are bits and P = Stop. Image (right) from Wikipedia. (https://en.wikipedia.org/wiki/I%C2%B2C)*

SPI or serial peripheral interface is a synchronous digital communication interface for short distances and primarily used in embedded systems. The standard was invented by Motorola and first introduced as an external microcontroller bus in 1979. The SPI bus includes four logic signals, serial clock (SCLK), master out/slave in (MOSI), master in/slave out (MISO) and slave select (SS). The clock polarity and phase can be programmatically configured to four different SPI modes (i.e. 0, 1, 2 and 3). Figure 3 displays an SPI implementation with three slaves (left), the four SPI modes (middle) and a corresponding sample timing plot for the respective four SPI modes (right). Alternative naming conventions is widely

used for these signals, but the functionality is the same. Contrary to the addressing schemes for the I²C and MIPI I3C standards, the SPI master uses the dedicated slave select (SS) line to address each slave individually. The two extra lines required for SPI puts it at a disadvantage compared to the two-wire I²C/MIPI I3C interfaces in applications where real-estate is precious and where the master has limited digital output channels for multiple SS signals. The SPI interface is push-pull (as opposed to I2C being open drain) and supports high-speed bitrates up to 100 MHz. There is no limitation for the number of bits per transfer (as opposed to I²C that transmits 8 bits per cycle) and it allows streaming of larger amounts of data. SPI is considered lower power than I²C since it requires less supporting circuitry and eliminates the pull-up resistors used for I²C.



SPI Mode	Clock Polarity (CPOL/CKP)	Clock Edge (CKE/NCPHA)
0	0	1
1	0	0
2	1	1
3	1	0

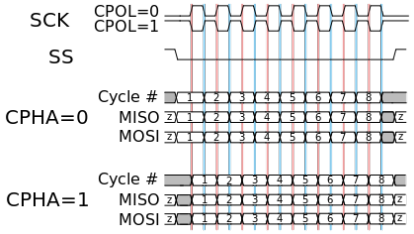


Figure 3: *SPI bus single master and three slaves (left), SPI modes (middle) and sample timing/mode diagram (right). Images from Wikipedia.*
(https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus)

Both SPI and I²C are legacy interfaces that have failed to evolve along with the market needs for lower-power, increased flexibility and interface standardization for sensor applications. They are widely used and accepted across many applications in many industries. While it would be tempting to develop a completely new, revolutionary digital communication interface, the evolutionary MIPI I3C interface allows co-existence between I²C and MIPI I3C devices. This enables a gradual migration to MIPI I3C and does not require an immediate upgrade of existing products (i.e. sensors) from I²C to MIPI I3C. The MIPI I3C interface strives to re-use, build upon and improve I²C for easy system integration while maintaining backward compatibility.

The main focus for the MIPI I3C standard has been to unify a fragmented interface industry by defining a standard that is low cost (i.e. takes up very little circuit real-estate on the sensor side), high-speed and that uses as little power as possible. The MIPI I3C standard provides immediate benefits for existing sensor systems and there are also provisions in anticipation for future mobile and other system architectures. The first MIPI I3C compliant IP blocks were announced in April 2016 with immediate availability. These IP blocks can be dropped directly into new IC designs to add the MIPI I3C sensor interface. This solution supports all MIPI I3C data rates up to 33.3 Mbps, dynamic address allocation and multi-master operations. It is also compliant with the MIPI Camera Control Interface (CCI) and backward compatible with I²C. The idea is meant for developers to future proof their designs for legacy I²C devices while utilizing the new functionalities that the MIPI I3C standard offers. Table 2 compares and contrasts MIPI I3C, I²C and SPI.

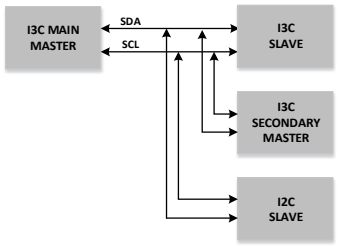
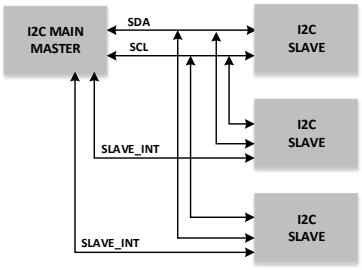
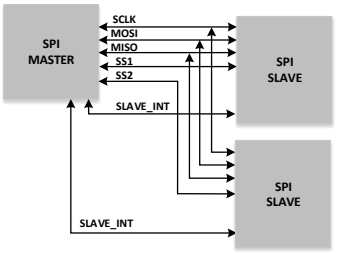
Parameter	MIPI I3C (Improved Inter Integrated Circuits)	I²C (Inter Integrated Circuits)	SPI (Serial Peripheral Interface)
Overview			
Number of Lines	2-wire	2-wire (plus separate wires for each required interrupt signal)	4-wire (plus separate wires for each required interrupt signal)
Effective Data Bitrate	33.3 Mbps max at 12.5 MHz (Typically: 10.6 Mbps at 12 MHz SDR)	3 Mbps max at 3.4 MHz (Hs) 0.8 Mbps max at 1 MHz (Fm+) 0.35 Mbps max at 400 KHz (Fm)	Approx. 60 Mbps max at 60 MHz for conventional implementations (Typically: 10 Mbps at 10 MHz)
Advantages	• Only two signal lines • Legacy I²C devices co-exist on the same bus (with some limitations) • Dynamic addressing and supports static addressing for legacy I²C devices • I²C-like data rate messaging (SDR) • Optional high data rate messaging modes (HDR) • Multi-drop capability and dynamic addressing avoids collisions • Multi-master capability • In-band Interrupt support • Hot-join support • A clear master ownership and handover mechanism is defined • In-band integrated commands (CCC) Support	• Only two signal lines • Flexible data transmission rates • Each device on the bus is independently addressable • Devices have a simple master/slave relationship • Simple implementation • Widely adopted in sensor applications and beyond • Supports multi-master and multi-drop capability features	• Full duplex communication • Push-pull drivers • Good signal integrity and high speed below 20MHz (higher speed are challenging) • Higher throughput than I²C and SMBus • Not limited to 8-bit words • Arbitrary choice of message size, content and purpose • Simple hardware interfacing • Lower power than I²C • No arbitration or associated failure modes • Slaves use the master's clock • Slaves do not need a unique address • Not limited by a standard to any maximum clock speed (can vary between SPI devices)
Disadvantages	• Only 7-bits are available for device addressing • Slower than SPI (i.e. 20Mbps) • New standard, adoption needs to be proven • Limited number of devices on a bus to around a dozen devices	• Only 7-bits (or 10-bits) are available for static device addressing • Limited communication speed rates and many devices do not support the higher speeds • Slaves can hang the bus; will require system restart • Slower devices can delay the operation of faster speed devices • Uses more power than SPI • Limited number of devices on a bus to around a dozen devices • No clear master ownership and handover mechanism. • Requires separate support signals for interrupts	• Need more pins than I²C/MIPI I3C • Need dedicated pin per slave for slave select (SS) • No in-band addressing • No slave hardware flow control • No hardware slave acknowledgment • Supports only one master device • No error-checking protocol is defined • No formal standard, validating conformance is not possible • SPI does not support hot swapping • Requires separate support signals for interrupts

Table 2: *Comparison between digital communication interfaces: MIPI I3C, I²C and SPI.*

MIPI I3C Protocol and System Integration

The MIPI I3C interface provides several communication protocols including an I²C like single data rate (SDR) messaging mode running up to 12.5 MHz and several high data rate (HDR) messaging modes that are not I²C compatible. Both the SDR and HDR formats share a two-wire interface with a bidirectional data pin (traditionally SDA) and one pin either used as a clock pin (SCL on SDR and HDR-DDR protocols) or as a bidirectional data pin (on HDR-TSL and HDR-TSP protocols). The I²C compatible SDR format supports a mix of various message types like standard I²C messages, broadcast and CCC messages that allow the master to communicate to all devices on the bus and slave initiated requests (e.g. in-band interrupts or requests to assume the master role). The SDR mode is more flexible than I²C and allows a secondary master and a mix of I²C and MIPI I3C slaves. I²C slaves must be accommodated by standard I²C data rate speeds and communication protocol initiated by a start-bit followed by a 7-bit slave address. In other words, the MIPI I3C SDR mode could be operated exactly like an I²C bus. All devices on the bus must be upgraded to MIPI I3C to be able to take full advantage of the MIPI I3C interface like the HDR modes, though.

The high speed MIPI I3C modes support speeds beyond the 12.5 Mbps base SDR mode. There are two main HDR modes, HDR-DDR (double data rate) and HDR-TSL/TSP (ternary symbol). These modes offer bitrates over 33 Mbps at a fraction of the per bit power of I²C (400 kHz, fast mode). Slave-side implementation is simple and this mode can coexist with legacy I²C devices. In practice, this means that HDR-DDR can be used to communicate with MIPI I3C slaves while allowing I²C devices to be on the same bus and communicate with these using legacy I²C interface. The I²C devices are expected to simply ignore the high speed MIPI I3C HDR transmissions. The HDR-DDR mode uses the SCL signal as a clock with data bits being clocked on both SCL edges (I²C and MIPI I3C SDR modes change SDA only when SCL is low). HDR-DDR moves data by "words", which contain two preamble bits, 16 payload bits and two parity bits. There are four word types defined: command word, user data word, cyclic-redundancy-check (CRC) word and reserved word. The different word types have specific functions, like the command word that indicates the direction of data movement either as write (master to slave) or read (slave to master). The HDR-TSL/TSP enables ternary (i.e. base three numerals codes) symbol coding for pure MIPI I3C implementation (TSP) and I²C-legacy inclusive (TSL) systems. HDR-TSL is using both SCL and SDA as data lines, where at least one line must transition each period. Transition indices are used for encoding binary into ternary symbols transfer to enable high speed transmission at very low power.

A major challenge for smartphones, wearables and other systems is to integrate multiple sensors that are using different communication protocols. Similar sensors from various vendors have been notoriously inconsistent in specifying sensor performance and communication interface. Back in 2013, Intel, Qualcomm and MSIG pioneered a sensor performance standard for mobile devices called the "Standardized sensor performance parameter definitions"[4]. This later became the IEEE 2700 standard. Similarly there is a need to standardize how microcontrollers and application processors communicate with sensors in a system. Currently, each sensor has its own set of functions and command structure. It would be highly beneficial to have a unified "sensor link" that would use a consistent implementation for all types of sensors. As an example, a "Read Pressure" command for a barometric pressure sensor would return a 16-bit pressure result independent of its vendor. This sensor link could allow pre-

[4] "Standardized sensor performance parameter definitions", MEMS Industry Group, May 2014 http://www.memsindustrygroup.org/default.asp?page=SPD

compiled libraries that would greatly simplify software development for the transfer layer between the master and slaves on the MIPI I3C bus.

Case Study | MIPI I3C for smartphones

The many sensors packed into modern smartphones are enabling advanced features like activity recognition, pedestrian navigation, health and fitness tracking capabilities and others. Smartphones are employing sensor fusion and algorithms to recognize the difference between walking and driving, or to make clever power saving schemes like switching off the Wi-Fi if the phone has been idle for a period of time. As the trend for adding more sensors continues, the implementation is quickly becoming unmanageable. High-end smartphones already have 10 sensors or more, and can require up to 20 signal lines. Additional sensors will require additional logic lines and increase the overall power consumption. Then there are the always-on features that enable constant monitoring of sensor functions even when a device is in idle mode. A smartphone located in the user's pocket or purse will continue to run the pedometer mode for step counting, along with other activity recognition features. The fact that some sensors are always active and that sensor data is always being transferred between devices, requires a very low power communication interface. Both I²C and SPI are typically used to support multiple sensors, but they both have drawbacks for sensor interconnections. Neither of them have a method to notify the master about a change in state or to initiate a sensor data transfer. These notifications are currently being performed by external general purpose input and output (GPIO) signals. MIPI I3C can replace both I²C and SPI with a more power efficient two-wire interface. The MIPI I3C standardized sensor interface will eliminate or reduce the need for external GPIOs and substitute these with in-band interrupts. The result is a simpler and more flexible implementation. Table 3 lists a number of sensors and other devices that are being targeted by the MIPI I3C interface.

Mechanical / Motion	Environmental Sensing	Biometrics / Health	Other
• Compass / magnetometer • Gyroscope • Accelerometer • Proximity • Touch screen • Grip • Time of flight (gestures) • Audio / ultrasonic	• Ambient light • Barometric pressure / altimeter • Temperature • Carbon monoxide / pollutants • Humidity	• Fingerprint • Glucometer • Heart rate • Olfactory (smell) • EKG • GSR (galvanic skin response)	• NFC • Haptic feedback • IR (smart TV remote) • UV • RGB

Table 3: *Partial list of sensors and other functions targeted by the MIPI I3C interface.*

Case Study | MIPI I3C for wearables

Wearables are characteristically small and have severe power limitations. It would be highly desirable to replace the commonly used I²C bus with a more power efficient and flexible interface. Similar to smartphones, the MIPI I3C could replace digital control signals for sensors with in-band interrupts. This would save space on densely populated printed circuit boards (PCBs). The images in Figure 4 display a

simple wearable that only contains one sensor, an accelerometer. The wearable's electronic module is shown to scale (middle) along with the PCB that is mounted inside the electronic module (right). There is not a lot of room on this PCB for components and traces, nor is there much space for a battery inside the housing. Needless to say, the ability to both save real estate and reduce power are two very critical aspects for developing new and better performing wearables.

Figure 4: *Simple activity tracker Fitbit Flex, product (left), electronic module (middle) and PCB assembly with the accelerometer marked with a yellow square (right). Images from UserLib (http://www.userlib.com/fitbit-flex-manual-tutorial/) and iFixit (https://www.ifixit.com/Teardown/Fitbit+Flex+Teardown/16050).*

Case Study | MIPI I3C for IoT devices

Internet of Things (IoT) devices are making everyday items like homes and cars smarter to improve our daily lives. This would not be possible without the use of sensors to gather and analyze data from the world around us. Figure 5 shows a typical system with two types of sensor hubs that interface directly to a range of sensors. The sensor hubs act as the master and control the communication at all times, for both I2C and SPI. In this system, MIPI I3C can replace all other communication interfaces with two wires. It will allow slave devices to initiate communication with simple in-band interrupt requests.

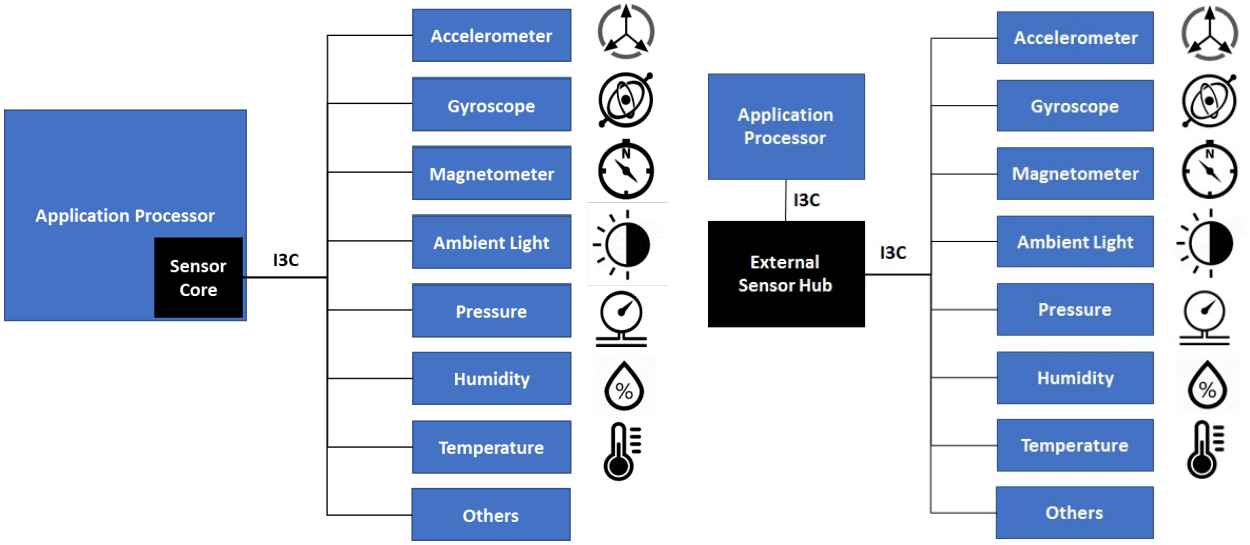


Figure 5: *Two common types of sensor hub architectures for mobile, wearables and IoT devices; application processor with sensor hub (left) and external sensor hub (right).*

Introduction to the MIPI I3C Standardized Sensor Interface

Summary

The MIPI I3C standardized sensor interface is a game changer for integrated sensor systems. It has built a superset of features on top of the existing I²C (two-wire) interface with additional high data rate modes that can satisfy sensor use cases that currently require an SPI bus (four-wire). MIPI I3C is an evolutionary standard that combines the advantage of I²C and SPI while adding new features such as in-band interrupts, dynamic addressing and advanced power management. It is defined to maintain backward compatibility with I²C and offers drastically lower cost, lower power and better scalability than I²C, SPI, UART and other digital interfaces. Sensor usage in modern smartphones that currently requires up to 20 signal lines including interrupts, can now be replaced by two. This is a major shift in streamlining sensor integration, which we expect to drive cost efficiencies and standardize a fragmented industry. MIPI I3C has clear advantages in mobile and consumer electronics like smartphones, tablets and wearables. It is also practical for other use cases that employ sensors, like IoT devices and applications for medical, industrial, automotive and others. With a widespread adoption, MIPI I3C has great potential and could extend to other non-sensor devices such as touch sensing, always-on, and low-resolution cameras (higher resolution cameras and displays are covered by other MIPI Alliance interfaces). If this happens, MIPI I3C could truly be a standard for the future.

Contact Information:
Melanie Cole (melanie.cole@mipi.org)
Manager, Integrated Marketing & Membership
MIPI Alliance (www.mipi.org)