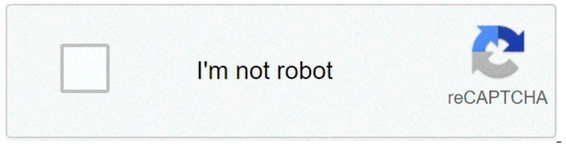

Continue

Difference between i2c and spi protocol

There is a lot of serial communication protocol but in which I2C and SPI are very famous, In this article, I will discuss the difference between I2C and SPI (I2C vs SPI). I2C and SPI both are bus protocol to allow the user for short-distance, serial data transfer. I2C is two-wire communication made by Philips (Nowadays NXP) and SPI is made by Motorola. Both protocols are commonly used in electronic devices like smartphones, TV, and laptops to control peripherals like power management chips, memory devices, input devices, etc. What is I2C? I2C is a serial communication protocol. It provides good support to the slow devices, for example, EEPROM, ADC, and RTC etc.I2c are not only used with the single board but also used with the other external components which have connected with boards through the cables. I2C is basically a two-wire communication protocol. It uses only two-wire for communication. In which one wire is used for the data (SDA) and other wire is used for the clock (SCL). In I2C, both buses are bidirectional, which means the master able to send and receive the data from the slave. The clock bus is controlled by the master but in some situations slave is also able to suppress the clock signal, but we will discuss it later. Additionally, an I2C bus is used in the various control architecture, for example, SMBus (System Management Bus), PMBus (Power Management Bus), IPMI (Intelligent Platform Management Interface), etc. Read the article to know I2C protocol in detail: Understanding of I2C Protocol. Advantages of I2C communication protocol There is a lot of advantage of I2C protocol which makes the user helpless to use the I2C protocol in many applications. It is the synchronous communication protocol, so no need of precise oscillators for the master and slave. It requires only two-wire, one wire for the data (SDA), and other wire for the clock (SCL). It provides the flexibility to the user to select the transmission rate as per the requirements. In I2C Bus, each device on the bus is independently addressable. It follows the master and slave relationships. It has the capability to handle multiple masters and multiple slaves on the I2C Bus. I2C has some important features like arbitration, clock synchronization, and clock stretching. I2C provides ACK/NACK (acknowledgment/ Not-acknowledgement) features that provide help in error handling. Some important limitation of I2C communication protocol An I2C protocol has a lot of advantage but beside it, I2C has a few limitations. It consumes more power than other serial communication busses due to open-drain topology. It is good only for a short distance. I2C protocol has some limitation for the number of slaves, the number of the slave depends on the capacitance of the I2C bus. It only provides a few limited communication speed like 100 kbit/s,400 kbit/s, etc. In I2c, devices can set their communication speed, slower operational devices can delay the operation of faster speed devices. What is SPI? The serial peripheral interface is four wire-based full-duplex communication protocol these wire generally known as MOSI (master out slave in), MISO (master in slave out), SCL (a serial clock which produces by the master) and SS (slave select line which use to select specific slave during the communication). SPI follows the master and slave architecture and communication is always started by the master. Like I2C it is also a synchronous communication protocol because the clock is shared by the master and slave. SPI is supported only multi-slave does not support multi-master and slaves are selected by the slave select signal. In SPI during the communication data is shifted out from the master and shifted into the slave vice- versa through the shift register. Advantages of SPI communication protocol There is no start and stop bits, so the data can be streamed continuously without interruption. It supports full-duplex. No need for precision oscillators in slave devices as it uses a master's clock. No complicated slave addressing system like I2C. Higher data transfer rate than I2C (almost twice as fast). Separate MISO and MOSI lines, so data can be sent and received at the same time. Simple software implementation. Disadvantages of SPI communication protocol If there is more than one slave in communication then the wiring will be complex. Uses four wires (I2C and UARTs use two). No acknowledgment that the data has been successfully received (I2C has this). No form of error checking like the parity bit in UART. It only allows for a single master. If you want to learn STM32 from scratch, you should follow this course "Mastering Microcontroller with Embedded Driver Development". The course contains video lectures of 18.5-hours length covering all topics like, Microcontroller & Peripheral Driver Development for STM32 GPIO, I2C, SPI, USART using Embedded C. Enroll In Course In the embedded system, I2C and SPI both play an important role. Both communication protocols are the example of synchronous communication but still, both have some important differences. In the below table, I have pointed out some common differences between SPI and I2C (SPI vs I2C). The important difference between I2C and SPI (I2C vs SPI) communication protocol. I2C can be multi-master and multi-slave, which means there can be more than one master and slave attached to the I2C bus. SPI can be multi-save but does not a multi-master serial protocol, which means there can be only one master attached to the SPI bus. I2C is a half-duplex communication protocol. SPI is a full-duplex commination protocol. I2C has the feature of clock stretching, which means if the slave cannot able to send fast data as fast enough then it suppresses the clock to stop the communication. Clock stretching is not the feature of SPI. I2C is used only two wire for the communication, one wire is used for the data and the second wire is used for the clock. SPI needs three or four-wire for communication ((depends on requirement), MOSI, MISO, SCL, and Chip-select pin. I2C is slower than SPI. In comparison to I2C, SPI is faster. I2C draws more power than SPI. Draws less power as compared to I2C. I2C is less susceptible to noise than SPI. SPI is more susceptible to noise than I2C. I2C is cheaper to implement than the SPI communication protocol. Costly as compared to I2C. I2C work on wire and logic and it has a pull-up resistor. There is no requirement of a pull-up resistor in the case of the SPI. In I2C communication we get the acknowledgment bit after each byte. Acknowledgment bit is not supported by the SPI communication protocol. I2C ensures that the data sent is received by the slave device. SPI does not verify that data is received correctly or not. I2C support multi-master communication. SPI does not support multi-master communication. I2C is a multi-master communication protocol that's why it has the feature of arbitration. SPI is not a multi-master communication protocol, so it does not consist of the properties of arbitration. I2C is the address base bus protocol, you have to send the address of the slave for the communication. In the case of the SPI, you have to select the slave using the slave select pin for the communication. I2C has some extra overhead due to start and stop bits. SPI does not have a start and stop bits. I2C supports multiple devices on the same bus without any additional select lines (work on the basis of device address). SPI requires additional signal (slave select lines) lines to manage multiple devices on the same bus. I2C is better for long-distance. SPI is better for a short distance. I2C is developed by NXP. SPI is developed by Motorola. Serial communications Help me by sharing this post PREVIOUS TUTORIAL NEXT PART In this page and video you will see the main differences between 3 protocols of serial communication, UART, i2c and SPI. Serial communication is the most widely used communication methodology as far as embedded systems are concerned. Before talking about types of serial protocols used in embedded industry and comparing them, let us first see what a serial communication is. PART 1 - What is Serial Communication? As its name suggests, in this kind of communication data is transferred serially (one after another) and not parallel (everything together). So as expected, a serial communication can be done using fewer wires as compared to its parallel counterpart and it also needs some sort of syncing mechanism (clock) to make a successful communication. In Serial Communication, only communication is done serially rest everything, like processing of the data etc., happens in a parallel fashion i.e. in form of registers. Serial communication can be further categorized into synchronous and asynchronous type. Synchronous serial communication: In this type of communication both transmitter and receiver share a common clock to remain in sync with each other. Asynchronous serial communication: This type of serial communication does not require any common clock source between the transmitter and receiver, both the sides work according to their independent clocks. Types of Serial Communication Protocols: There are various types of serial communication protocols being used in embedded industry. Let us discuss them here: PART 2 - SPI (Serial Peripheral Interface) This is a synchronous type serial communication protocol which consists of two data lines (MOSI and MISO), one clock line (SCK) and a slave select line (SS). Before moving ahead here are some terms that you should be aware of: Master – Device which provides clock for communication Slave – Device other than master which utilises master's clock to communicate MOSI – Master Out Salve In (line though which master sends data to its slaves) MISO – Master In Slave Out (line though which salves responds back to the master) SCK – Serial Clock (clock provided by master device) SS – Slave Select (line used to select slave to which master wants to communicate) In a SPI, at any given time there could be only one master device and several other slaves under it who only respond to master's call. The entire communication is handled by master itself; no slave can send data on its own will. Master sends data via MOSI while slaves respond via MISO line. In the entire process SCK (serial clock) plays a very important role, every slave device depends on this clock to read data from MOSI and respond through MISO. SS (slave select) is used to make a particular slave awake with who master wants to communicate. Here is an illustration of SPI: Now there are few register which are used to implement SPI communication. We have these below and as you can se we have SPDR, SPSR and SPCR so let's see each one. SPDR (SPI Data Register) ➢ This is used to store one byte of data which is to be transferred or received. SPSR (SPI Status Register) ➢ This register holds the status bits involved in SPI communication SPCR (SPI Control Register) ➢ This register holds the control bits involved in SPI communication. All the above registers are 8 bit in length. Advantages: 1. Provides synchronous serial communication which is much more reliable over asynchronous 2. Multiple devices(Slaves) can be connected to single master 3. Faster form of serial communication Disadvantages: 1. Requires multiple slave select wires for connecting multiple slaves 2. Only master has control over entire communication process; no two slaves can communicate with each other directly PART 3 - I2C (Inter-Integrated Circuit) or Two Wire Interface Another very useful synchronous serial communication protocol is I2C or Inter-Integrated Circuit protocol. Unlike SPI, I2C uses only two wires for the entire process, maybe that's why it is also known as Two Wire Interface (TWI) protocol. These two wires are SDA (Serial Data) and SCL (Serial Clock). I2C protocol can support multiple slave devices but unlike SPI, which only supports one master device, I2C can support multiple master devices as well. Every device sends/receives data using only one wire which is SDA. SCL maintains sync between devices through common clock which is provided by the active master. Each slave has its own unique 7 to 10 bit address which master uses to identify them. Whenever master wants to send data it first generates a request which has particular address of that slave. Every slave matches this address with its own and the one whose address gets matched responds to the master. Every message initiates with a start condition and ends with a stop condition. A single message can hold multiple data bytes, each having an acknowledge (ACK) or negative acknowledge (NACK) bit in between them. Pull-up resistors with SDA and SCL are necessary in order to run this protocol. Advantages: 1. Multiple masters and multiple slaves can be interfaced together 2. Only two wires are required for this communication Disadvantages: 1. It is slower as compared to SPI because a lot of framing work is done within this protocol PART 4 - UART/USART UART stands for Universal Asynchronous Receiver and Transmitter while USART stands for Universal Synchronous and Asynchronous Receiver and Transmitter. The difference between them is that UART performs only asynchronous serial communication while USART can perform both synchronous as well as asynchronous serial communication process. For Asynchronous mode, this protocol makes use of only two wires i.e. Rx and TX. Since no clock is needed here, both the devices have to make use of their independent internal clocks to work. Yet there is a term called baud rate which helps these devices to remain in sync by fixing the speed of data exchange. Baud rate refers to the number of data bits transmitted per second, so both devices should work on same baud rate in order to maintain its proper functioning. UART/USART has a big limitation that only two devices can communicate using this protocol at once. TX pin of one device transmits data to the RX pin of another device and similarly TX of latter transmits data to RX of former device. This is how exchange of data takes place. *Note: Both the communicating devices should have a common ground (GND). Advantages: 1. Provides both synchronous as well as asynchronous serial communication 2. Availability of various baud rates making it suitable for wide applications and devices 3. One of the easiest form of serial communication Disadvantages: 1. Can connect only two devices at a time Conclusion Use SPI when you have only one master and multiple slave devices. SPI proves to be a faster protocol for this. When you have multiple master devices as well, apart from multiple slave devices, then one should prefer using I2C or TWI over SPI. This will also reduce the number of wires to be used. Now if you are looking for a device to device serial communication then USART/UART proves itself the best as it is easy to deal with and widely used in many peripheral devices. PREVIOUS TUTORIAL NEXT PART Help me by sharing this post