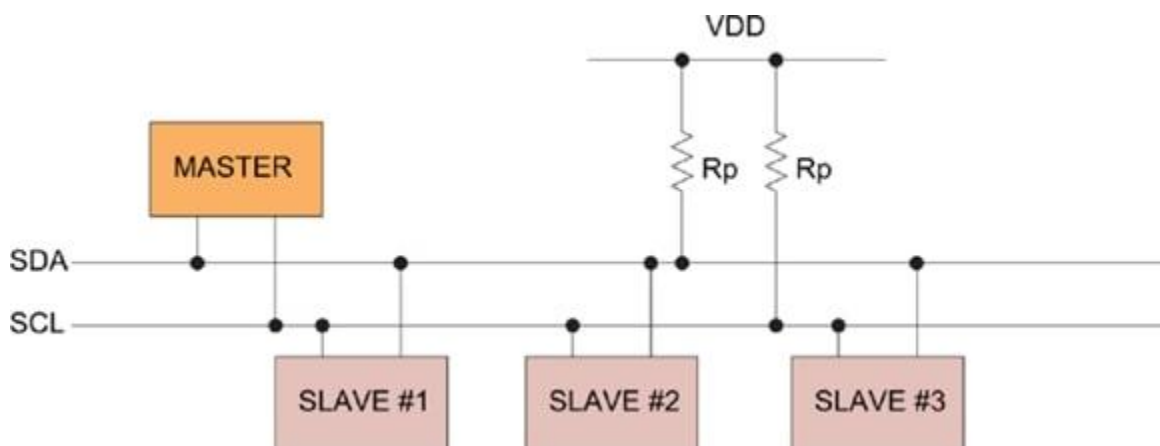


I2C Protocol

I2C Protocol

Introduction to I2C Protocol:



I2C stands for Inter-Integrated circuit. It can also be referred to as IIC or I2C. I2C Protocol is a serial communication protocol, and it is widely used for short-distance communication. It provides simple and robust communication between the Peripheral device and the microcontroller.

I2C Protocol consists of two wires SDA and SCL which is bidirectional synchronous serial bus communication. Thus, I2C Protocol takes two wires for communication which also translates to low cost and this low cost has made I2C Protocol the most commonly used Serial Bus across most applications including IoT, consumer electronics, automotive, aerospace, and industrial equipment.

I2C Protocol is a synchronous protocol that allows a master to initiate communication with a slave device. I2C protocol is a master-slave communication where the master provides the clock which becomes the data transfer rate or clock frequency. It is a bi-directional bus meaning the master can write to the slave and read from the slave it is a serial bus which means data is a clock and it is shifted bit by bit and there are two bus lines serial clock (SCL) and serial data (SDA).

As you see in the diagram three slaves are connected to the same I2C Master and there is two pull-up resistor which is required for the I2C device to communicate properly this is because the I2C protocol works on the SCL and SDA bus lines which are an open drain or open collector the transmitting device just lets go of the I2C bus to create logic one and pulls or drives the line the ground to create logic 0.

In I2C Protocol there are 5-speed categories including standard mode, fast mode, fast plus mode, high-speed mode, and ultra speed mode these speed categories range from 100 kHz to 5MHz, where standard mode is 100KHz, Fast mode is 400KHz, Fast mode plus is 1MHz, High-Speed Mode is 3.4MHz and Ultra-fast mode is 5MHz. 100KHz up to 1MHz are very similar, while 3.4 MHz needs some special consideration, and 5MHz being unidirectional requires even more special attention. But the most common speed categories for I2C Protocol are Standard mode, Fast Mode, and Fast mode plus these modes are easy to implement.

I2C Protocol transactions are initiated with a start condition, the start condition is defined as the master driving SDA line low while SCL remains high, and note that it must be initiated with the [I2C bus](#) in an idle state means SDA and SCL lines are both high.

The slave address immediately follows the start condition, and it will be 7 bits or 10 bits long it should be noted that each device on the I2C bus needs a unique slave address. Next is the read-and-write bit, which immediately follows the slave's address, this bit informs the slave if the master wants to read or write. 1 indicates the master wants to read from the slave and 0 indicates the master wants to write.

The next section is the Acknowledge bit and we can think of ack bits like a handshake between the master and slave and they occur on every 9th clock cycle, if ACK is 0 then data can be transmitted, and if ACK is 1 data cannot be sent. Next is the data byte which contains the information being transferred between the master and slave, It contains 8 bits of data.

The last section of the I2C protocol is the Stop condition all I2C transactions should be terminated with a stop condition, the stop condition is defined as the master releasing the line while the SCL signal level is high after a stop condition the I2C bus will remain in idle state and won't be free for the next I2C transaction.

There are two important concepts of I2C Protocol: I2C Arbitration and [I2C Clock Stretching](#)

I2C Arbitration: I2C Protocol is a multi-master bus it can be possible that multiple masters are driving the bus simultaneously, If one of them detects SDA is low when it should be high, it assumes that another master is active and immediately stop its transfer.

I2C Clock Stretching: The master wants to send the data where the slave is not ready to receive data so the I2c slave pulls the SCL line low.

Advantages:

- I2C Protocol supports multi-master, multi-slave communication
- It uses only two wires
- In I2C Protocol ACK/NACK functions can be proved useful for error handling
- Adaptable as it can adapt to the needs of various slave device

- In I2C Protocol the addressing is very simple and does not need any CS lines to add extra devices like [SPI Protocol](#)
- It uses flow control

Disadvantages:

- I2C Protocol is a half-duplex mode of communication
- The hardware complexity increases when the number of master/slave devices is more in the circuit
- Many devices have multiple addresses stored which can cause conflicts

Applications of I2C Protocol:

- Accessing DACs and ADCs
- Reading certain memory ICs
- Reading hardware sensors
- Communicating with multiple micro-controller
- Transmitting and controlling user-directed actions

Debug I2C Protocol:

Prodigy also offers an [I2C Protocol analyzer](#) to capture and decode the I2C packets. The I2C protocol analyzer has the advanced capability not only to trigger but long allow very long captures which is very useful for the embedded design engineer during the debugging.