I2C(IIC,TWI)★2 >

I2C 통신 프로토콜

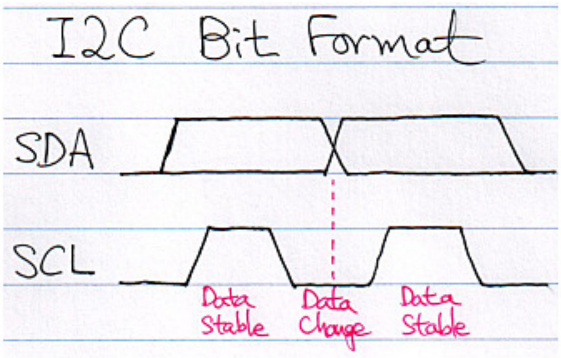
WKim 카페스탭 1:1 채팅
2015.10.27. 12:59 조회 1,324

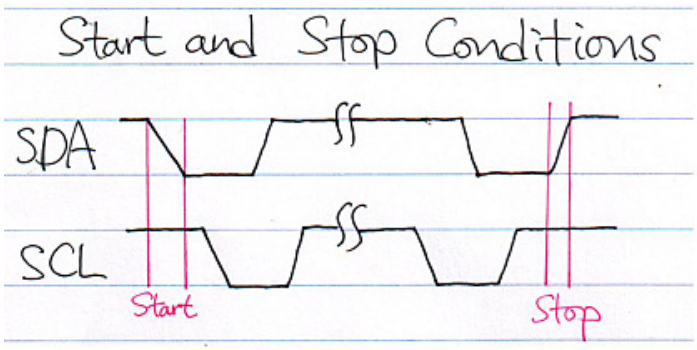
I2C(IIC, TWI) 통신 프로토콜(protocol)에 대해 알아보겠습니다.

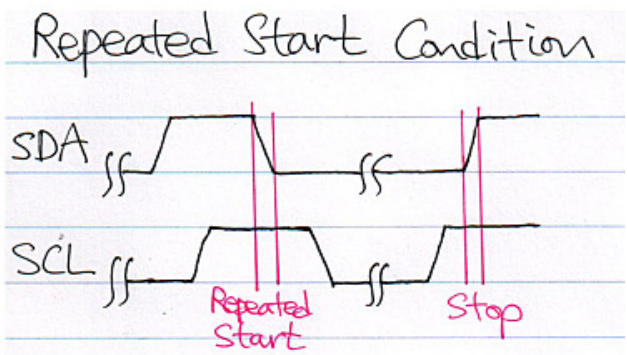
I2C 버스는 SDA(Serial Data)와 SCL(Serial Clock) 두개의 선호선으로 구성되어 있습니다. 최대 127개의 장치를 단 두개의 신호선으로 연결할 수 있는것이 장점이고, 최대속도는 400kHz입니다. SDA와 SCL선 모두 풀업저항이 걸려있고, 오픈드레인 또는 오픈컬렉터 출력을 하는 장치끼리 wired-AND형태로 연결되어 있어요. 따라서 어떤 장치가 low출력을 하면 신호라인의 상태는 low가 됩니다. I2C 버스에 master 장치 여러개를 연결할 수 있고, master 장치와 slave 장치 모두 데이터를 보낼 수 있습니다.

I2C 통신은 동기화(synchronous)된 시리얼 통신으로, SDA선으로 전달되는 비트신호는 SCL선으로 전달되는 클럭신호와 동기화되어 있어요. I2C bit format을 보면 SCL이 low일 때만 SDA신호가 변할 수 있고, SCL이 high일때는 SDA신호가 일정해야 합니다.

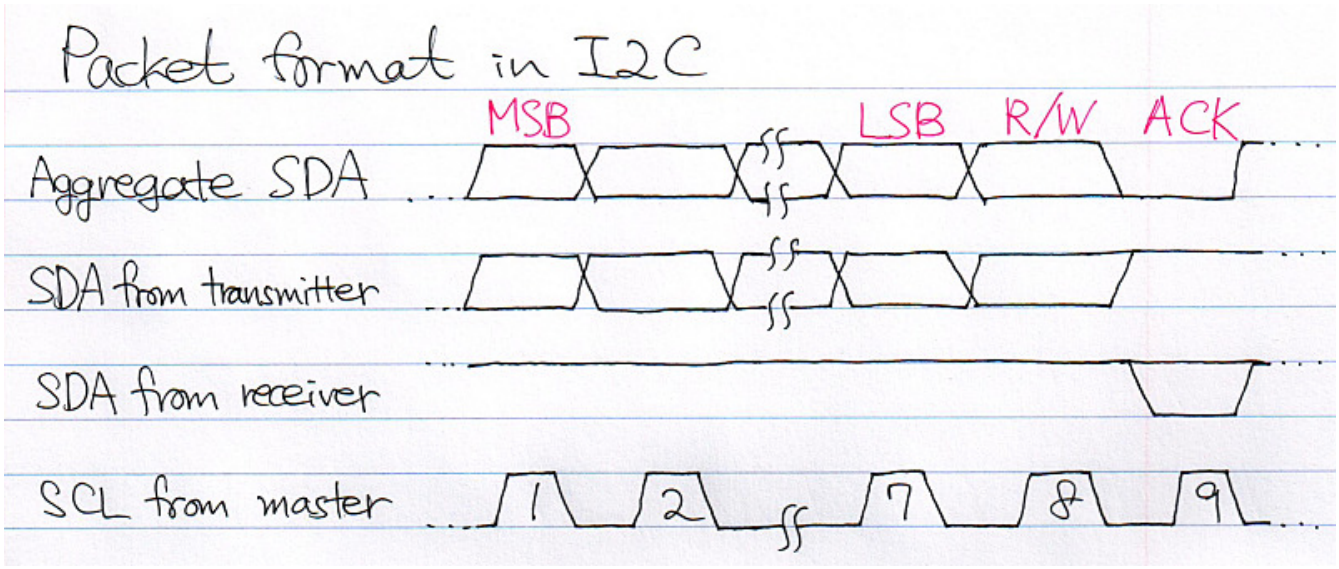
Start(시작), stop(끝) 신호는 여기서 예외인데요. 연속된 비트신호를 보내는 경우, SCL이 high일 때 SDA의 하강엣지가 나오면 이 신호의 시작을 나타냅니다. 반대로 SCL이 high일 때 SDA의 상승엣지가 나오면 이 신호의 끝을 나타내요. 연속된 비트신호의 끝을 나타내는 상승엣지가 나온 뒤 곧바로 하강엣지가 나오면, 다음 신호가 바로 시작하는 것을 나타냅니다. Start, stop 신호 사이에서는 I2C 버스가 사용중이라서 한 master 장치만 I2C 버스를 사용할 수 있어요.



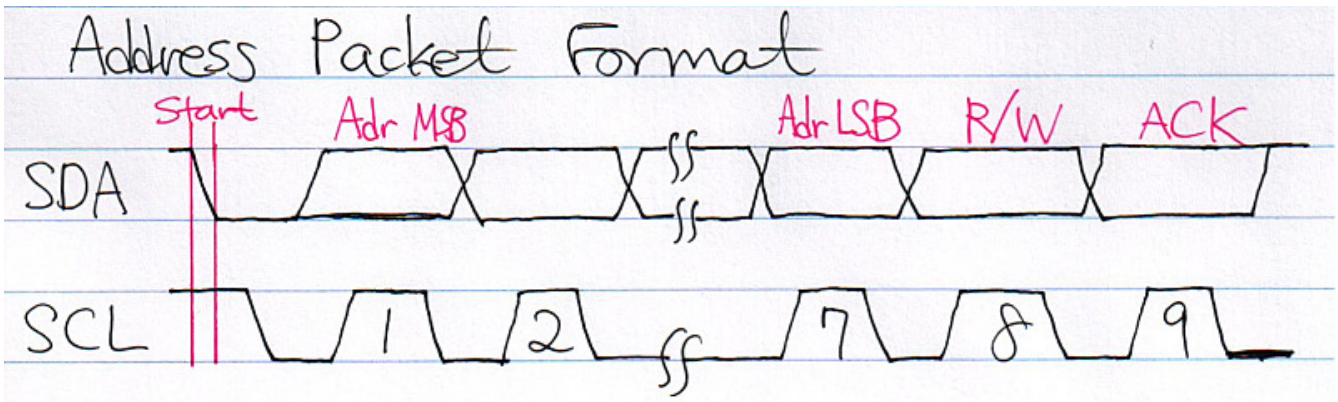




I2C 통신에서 주소 또는 데이터는 패킷의 형태로 전송되어야 합니다. 각 패킷은 9개 비트로 구성되는데, 처음 8비트는 transmitter에 의해 보내집니다. 마지막 9번째 비트는 receiver가 잘 받았다는 의미로 low신호, ACK(acknowledge)신호를 보냅니다. 만약 receiver가 마지막에 low신호를 보내지 않으면 SDA선이 high상태가 되고, transmitter는 이를 NACK(not acknowledge)으로 인식합니다. Transmitter가 NACK신호를 인식하면 방금 보낸 패킷을 다시 보내거나 패킷의 전송을 중단하는 등의 조치를 취해요.

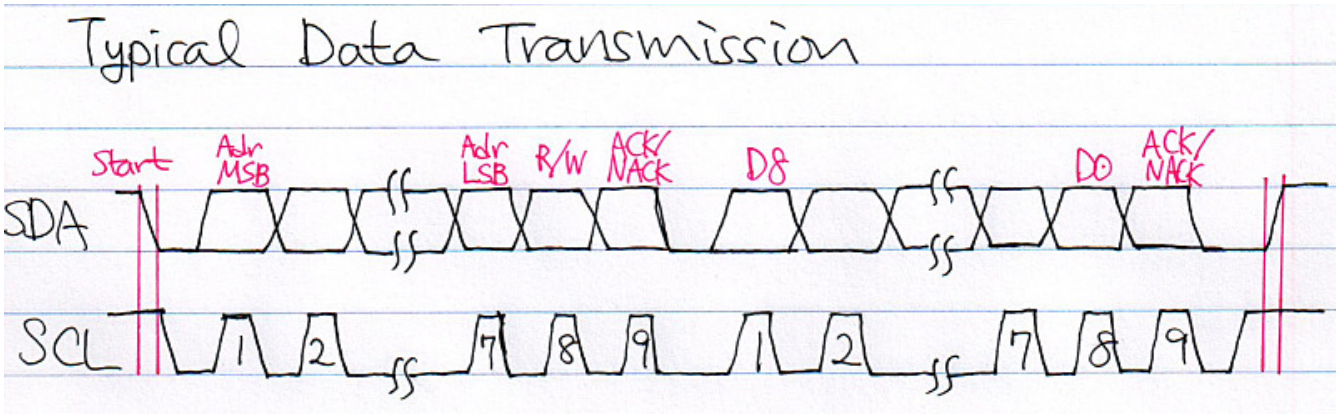


아래 그림은 I2C라인을 통해 주소 패킷을 전달하는 모습을 보여줍니다. Transmitter는 SCL이 high일 때 SDA라인을 내려서 전송의 시작을 알리고, 주소의 MSB부터 LSB까지 7개 비트를 보냅니다. 전송하는 주소가 7비트여서 128개의 주소가 가능하지만, 16개가 예약(reserverd)되었거나 다른 용도로 사용하게 되어있습니다. 그래서 7비트 주소를 사용하면 I2C 버스에 112개의 장치까지 연결할 수 있어요. 그 다음에 R/W(read/write) 여부를 알려주는 신호를 마지막으로 보내면, receiver가 ACK신호를 transmitter로 보냅니다.

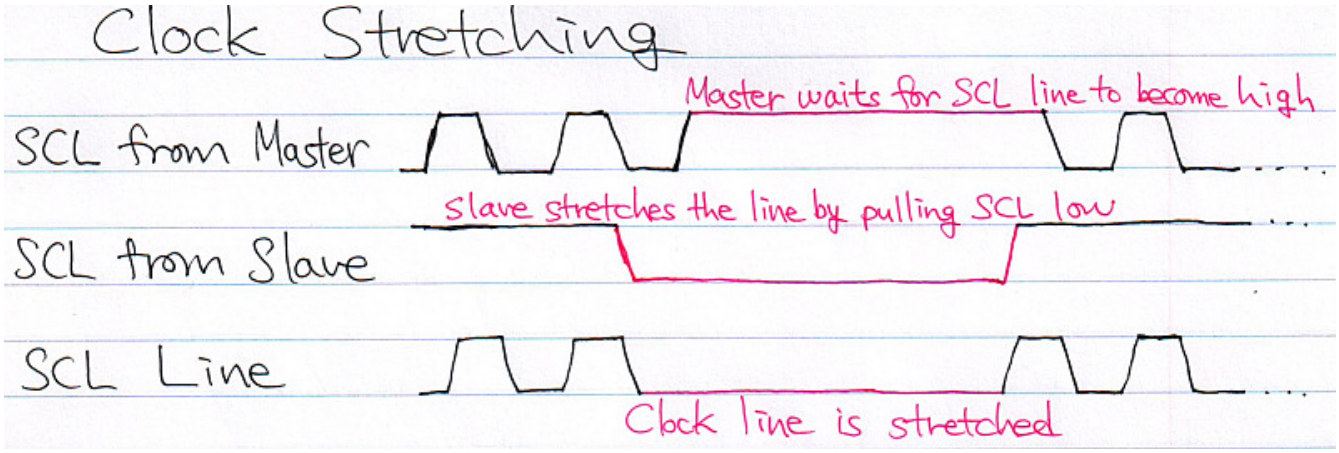


아래 그림은 주소패킷과 데이터패킷을 이어서 전송하는 것을 보여줍니다. 주소 패킷과 비슷한 방식으로 데이터 패킷을

보내는데, 주소패킷과 달리 데이터패킷은 8비트를 보냅니다.

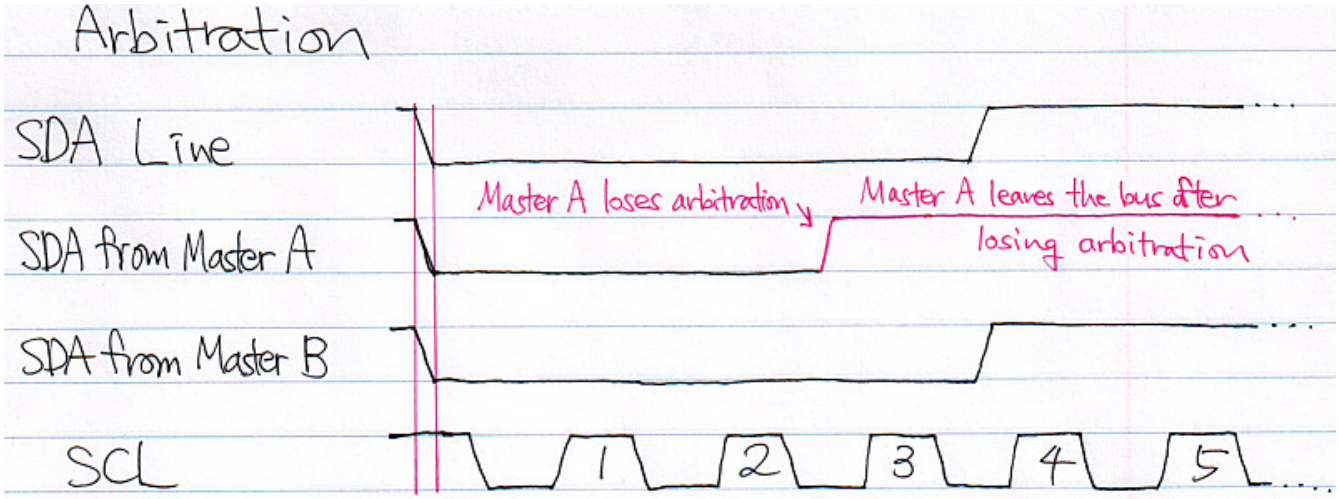


Slave 장치가 데이터를 받을 준비가 안되었을 경우, slave 장치는 SCL선의 상태를 low로 내려서 master 장치에게 데이터 전송을 미뤄달라는 요청을 할 수 있습니다. 이렇게 되면 master 장치는 slave 장치가 SCL선의 상태를 high로 올리기 전까지 SCL선으로 클럭신호를 발생시키지 않습니다. Slave장치가 SCL선을 high로 올린 뒤에 master장치는 SCL선으로 클럭신호를 발생시키는데, 이것을 clock stretching이라고 불러요.



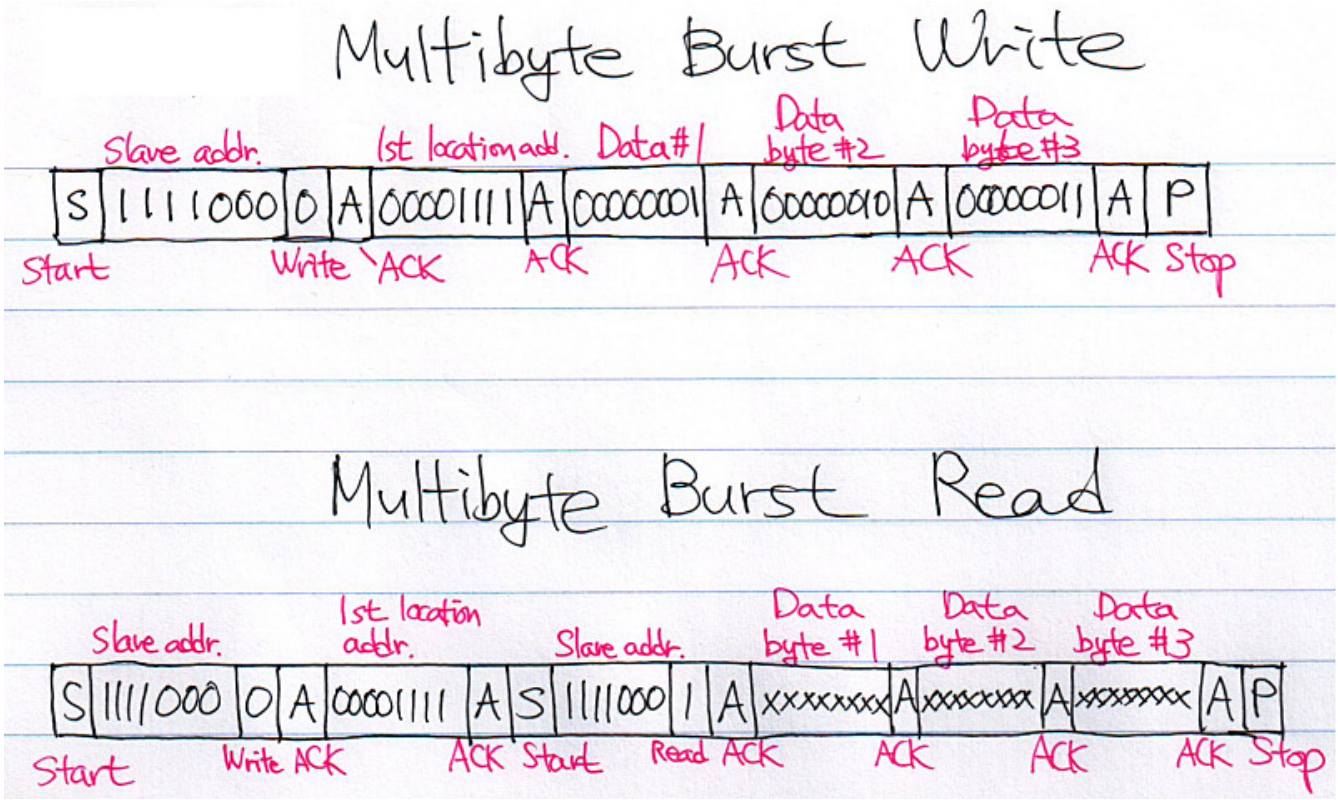
I2C 버스에 여러개의 master 장치를 연결할 수 있지만, 한 순간에는 단 하나의 마스터 장치만 I2C 버스를 사용할 수 있습니다. 만약 여러개의 master장치가 동시에 버스를 사용하려고 할 경우, arbitration이라는 일이 일어납니다. 각각의 master(transmitter)는 SDA라인으로 출력한 뒤 SDA라인의 상태가 실제로 의도한 상태인지를 체크하는데요. 만약 의도한 상태가 아니면 I2C 버스를 사용하는 것을 포기합니다.

아래에서 master A는 0010 000을 보내고, master B는 0001 111을 동시에 보내려고 하는데요. Master A가 세번째 클럭에서 high출력을 했지만 SDA선의 신호는 올라가지 않았습니다. 이 순간 Master A는 의도대로 출력이 되지 않음을 감지하고 I2C 버스의 사용을 포기해요. Master B가 경쟁에서 이겨서 I2C버스를 사용하게 됩니다.



연속된 데이터 주소에 효율적으로 값을 쓰기 위해, I2C통신이 burst mode로 동작하기도 합니다. Burst mode에서는 slave 장치의 주소, 데이터의 첫번째 주소를 보낸 뒤, 데이터를 바이트별로 보냅니다. 이 때 I2C장치는 stop신호가 나타날 때까지 데이터의 주소를 1씩 증가시킵니다. 연속된 데이터 주소의 값을 읽어올 때도 비슷한 방식으로 동작해요.

아래의 Multibyte Burst Write 예제는 slave 장치(주소: 1111000)의 0x0F부터 0x11까지 세 주소에 0x01, 0x02, 0x03의 데이터를 쓰게 합니다. 그 아래의 Multibyte Burst Read 예제는 slave 장치(주소: 1111000)의 0x0F부터 0x11까지 세 주소에 있는 값을 읽어오게 합니다.



손으로 직접 신호도를 그렸는데, 타이밍을 맞춰 그리기 조금 힘들었네요^^ I2C통신을 MCU로 다루기 전에 I2C 프로토콜을 알면 좋을 것 같습니다. 참고로 위 신호도는 다음 책에 나오는 내용입니다.

The AVR Microcontroller and Embedded Systems: Using Assembly and C, Muhammad Ali Mazidi, Pearson

댓글 등록순 최신순 댓글알림

키트

2015.10.27. 13:10 답글쓰기

제플린

2015.10.27. 13:18 답글쓰기

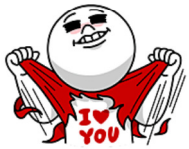
땅땅치킨a

2015.10.30. 06:18 답글쓰기

장난꾸러기

2015.11.03. 09:59 답글쓰기

아셀부비



2015.11.06. 18:10 답글쓰기

푸돌

2016.01.27. 17:54 답글쓰기

달비골

2016.02.03. 23:55 답글쓰기

신입연구원

2016.12.02. 14:01 답글쓰기

안양II제제

공부많이됩니다.감사합니다.

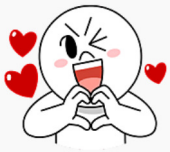
2016.12.29. 09:42 답글쓰기

이언섭

2017.06.08. 09:51 답글쓰기

pldworld

오~~ 눈에 쏙쏙 박히는 설명~
감사합니다~*^^*



2017.07.26. 15:24 답글쓰기

Register

좋은 글 올려주셔서 감사드립니다!!

2017.07.29. 00:39 답글쓰기

pldworld

위 내용중 clock stretching 부분을 설명하신 부분에서 다음의 문장이 있습니다.

"Slave장치가 SCL선을 low로 내린 뒤에 master장치는 SCL선으로 클럭신호를 발생시키는데, 이것을 clock stretching이라고 불러요."

아무래도 "Slave장치가 SCL선을 low로 내린 뒤에" 이 부분이 "Slave장치가 SCL선을 high로 올린 뒤에"라고 되어야 할 것 같습니다만...

2021.01.27. 14:54 답글쓰기

WKim 작성자

예, 언급하신 대로 수정하였습니다.

2021.01.27. 21:42 답글쓰기

pldworld

댓글을 남겨보세요

등록