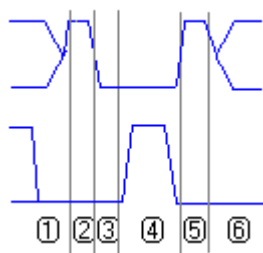


## I2C Bus Events: 슬레이브에 확인응답 (Acknowledge) 보내기

슬레이브로부터 바이트를 수신하는 즉시, 마스터는 이를 슬레이브 디바이스에 확인 응답해야 합니다. 마스터는 SDA 와 SCL 라인을 완전히 제어합니다.



마스터(1)로 마지막 비트의 전송 후에 슬레이브는 SDA 라인을 해제할 것입니다. SDA 라인은 그러면 high(2)로 가야 합니다.

마스터는 이제 SDA 라인을 low(3)로 끌어당길 것입니다. 다음으로, 마스터는 클럭 펄스를 SCL 라인(4)에 붙일 것입니다.

이 클럭 펄스의 완료 후, 마스터는 다시 SDA 라인(5)을 해제할 것입니다.

슬레이브는 이제 SDA 라인(6)의 제어를 되찾을 것입니다.

주의:

위의 파형은 다소 과장되었습니다. 여러분은 (2)와 (5)에서 SDA 가 high 로 가는 것을 알아차릴 수 없을 것입니다. 작은 스파이크는 거의 보이지 않을 것입니다.

알림:

수신된 최종 바이트 이외에는, 슬레이브로부터 수신된 바이트의 **Acknowledge** 는 항상 필요합니다.

마스터가 슬레이브로부터 데이터 수신을 멈추고 싶다면, **stop condition** 을 보낼 수 있어야만 합니다.

슬레이브는 마스터에 의해 발행된 **ACK** 사이클 후에 **SDA** 라인의 제어를 되찾게 되므로, 이것은 문제를 일으킬 수 있습니다.

마스터로 전송 준비된 다음 비트가 0 이라고 가정해 봅시다. 마스터가 SCL 라인을 low 로 취한 후에 SDA 라인은 슬레이브에 의해 즉시 low 로 당겨질 것입니다. 이제 마스터는 버스에서 Stop 조건 생성을 시도합니다. 이것은 먼저 SCL 라인을 해제하고 나서 SDA 라인 - 슬레이브에 의해 low 로 유지된 - 을 해제하려 합니다. 결론: 버스에서는 어떤 Stop 조건도 생성되지 않았습니다.

이 조건은 NACK 로 불립니다: Not ACKnowledge. 이것을 **No ACKnowledge** 와 혼동하지 마십시오:

상태	발생 가능...
Not acknowledge (NACK)	마스터가 슬레이브에서 바이트를 <b>read</b> 한 후
<u>No acknowledge</u>	마스터가 슬레이브에 바이트를 <b>write</b> 한 후