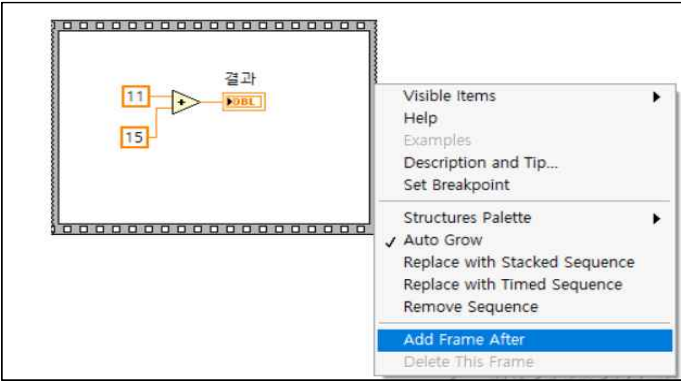
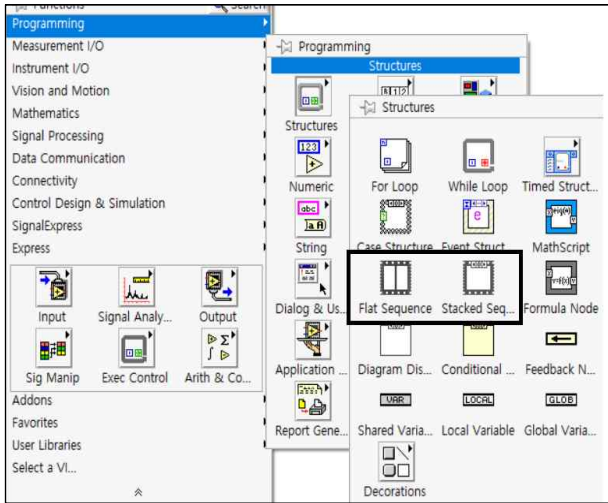
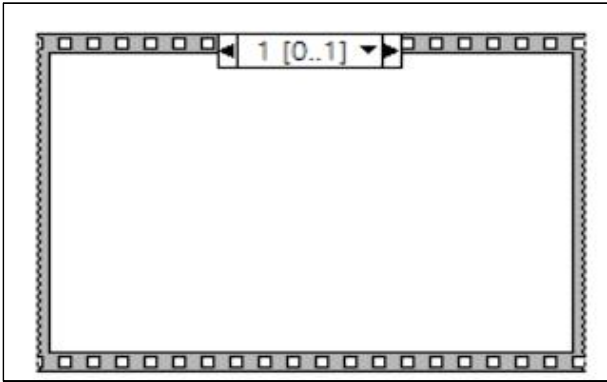
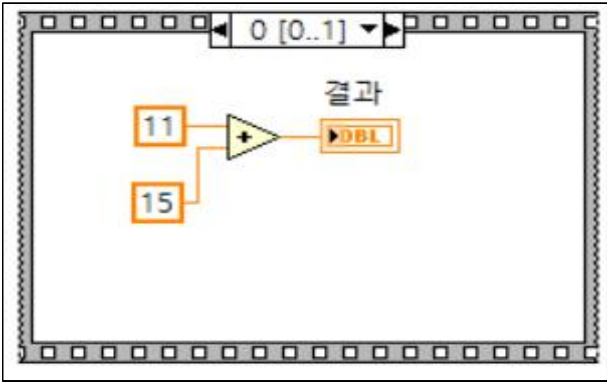


- 시퀀스 구조 : 시퀀스를 만들 때, 즉 순서를 강제적으로 정할 때 사용하는 구조.

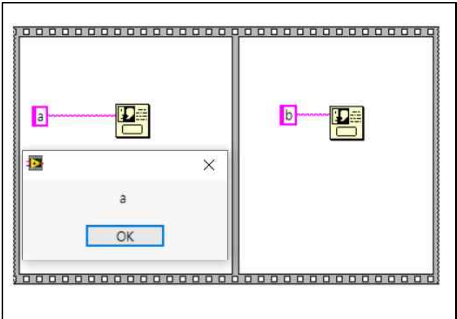
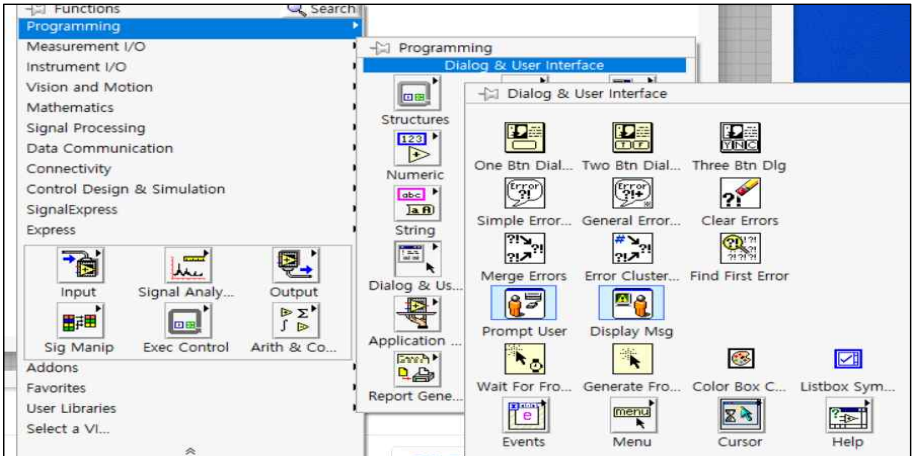


다음 프레임 추가

- 다층 시퀀스 구조를 사용하고 싶으면 플랫폼 시퀀스 구조를 사용하여 프로그램한 다음 플랫폼 시퀀스 구조의 “바로가기 메뉴 - ‘다층 시퀀스’로 대체”를 선택합니다.
- “바로가기 메뉴 - 다음 프레임 추가”를 이용하여 프레임을 추가할 수 있습니다.
- 플랫폼 시퀀스 구조는 **프레임이 다 펼쳐져 있어**, 코드를 한눈에 확인할 수 있어 좋긴 하지만, 공간을 많이 차지합니다.

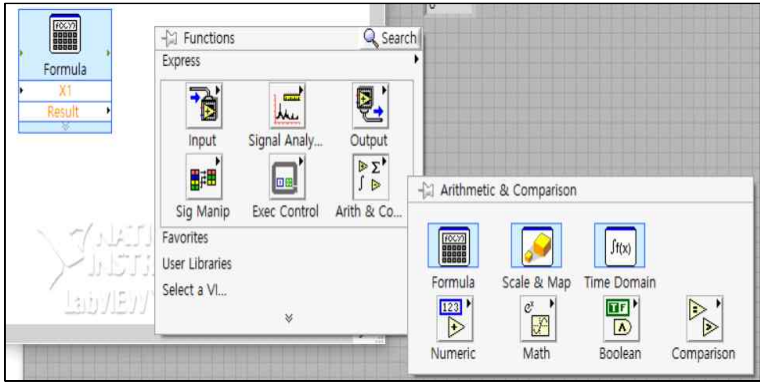


- 반면에 **다층 시퀀스 구조**는 프레임이 **차곡차곡** 쌓이게 됩니다. 다층 시퀀스 구조는 공간을 조금 차지하지만, 코드가 한눈에 보이지 않는 단점이 있습니다.
- 시퀀스 구조를 사용할 때는 각 프레임들을 실행하는 중에 에러가 발생하는 경우, 시퀀스를 중간에 빠져나올 수가 없습니다. 즉, 가지고 있는 모든 프레임을 다 실행해야지만, 시퀀스 구조를 빠져나가게 됩니다.

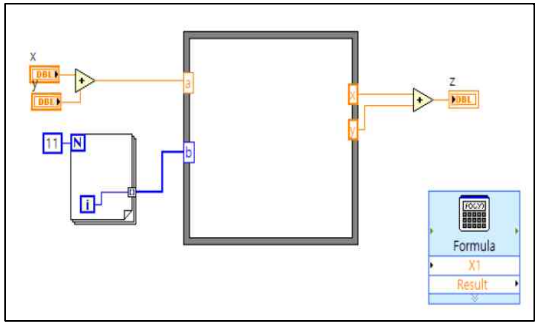
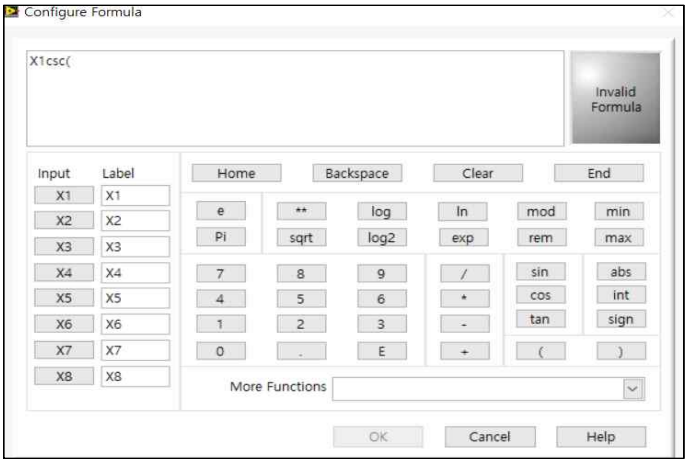


- one/two button dialog : 실행하면, ‘a’와 ‘b’의 대화창이 무작위로 나타납니다.

수식 노드(Formula node) : 복잡한 수식을 구현할 때, c언어와 비슷한 문법을 사용하여 구현이 가능합니다.

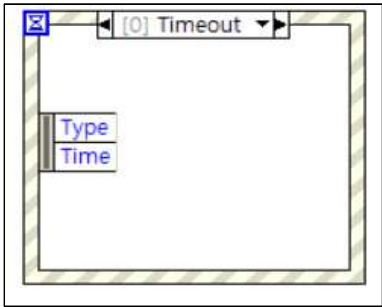


- 수식 노드와 비슷한 기능을 하는 노드 중 수식 익스프레스 노드가 있습니다.



이벤트 구조 : 프로그램 사용자가 프론트 패널에서 더블 클릭, 마우스 다운, 마우스 업, 키다운 등 마우스나 키보드를 통해 어떠한 동작을 할 때만 특정 코드를 실행하고자 할 때 이벤트 구조를 사용합니다.

- 정적(static) 이벤트(**다시 알림, 필터 이벤트**)와 다이내믹/사용자(**dynamic or user**) 이벤트로 구분됩니다.

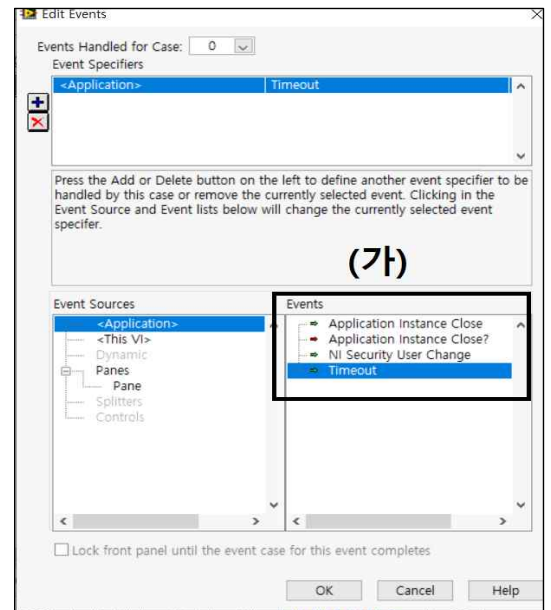
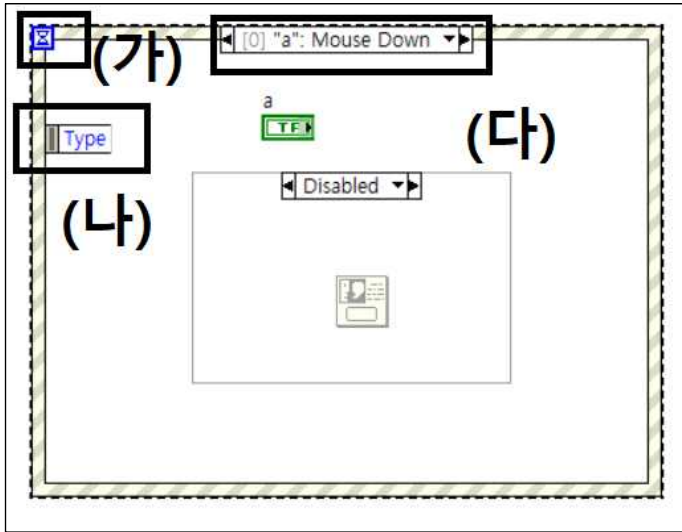


- 이벤트 선택자 라벨 : 어떤 이벤트인지를 구별
- 타임아웃 : 이벤트를 기다리는 시간 (msec)
- 이벤트 데이터 노드 : 이벤트가 가지고 있는 속성들
- 이벤트 필터 노드 : 이벤트가 가지고 있는 속성들.
- 다이내믹 이벤트 터미널 : 다이내믹 이벤트를 등록할 때 사용.

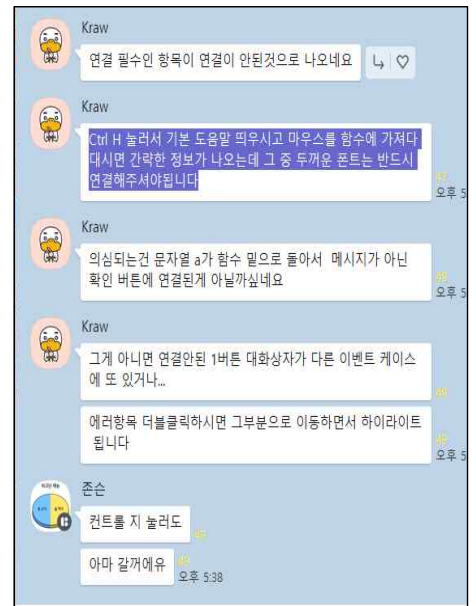
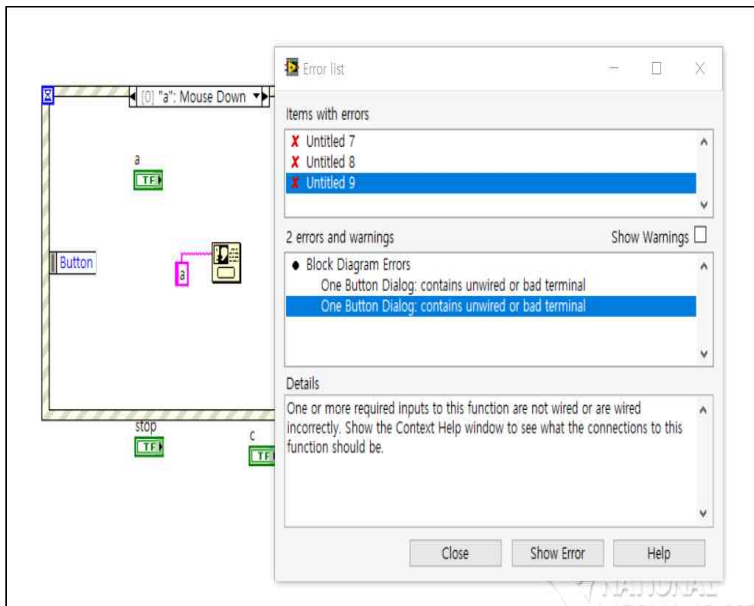
필터 이벤트 (가)

- **초록색 화살표** = 알림 이벤트로써 이벤트 중 가장 **빈번하게 사용**하게 될 이벤트
이벤트가 발생한 사실을 “Labview”에게 바로 알려 해당 특정 코드가 실행될 수 있도록 합니다.

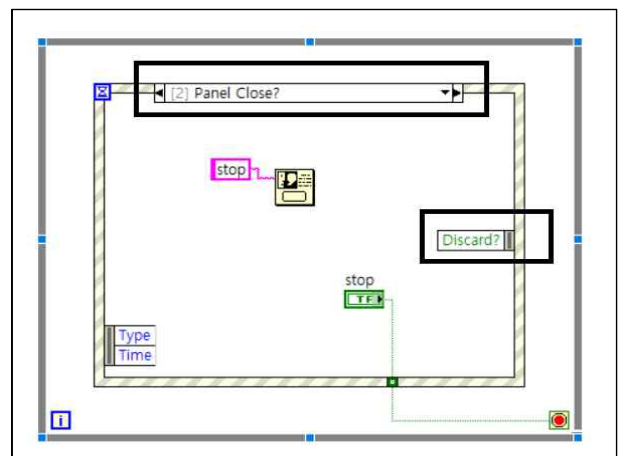
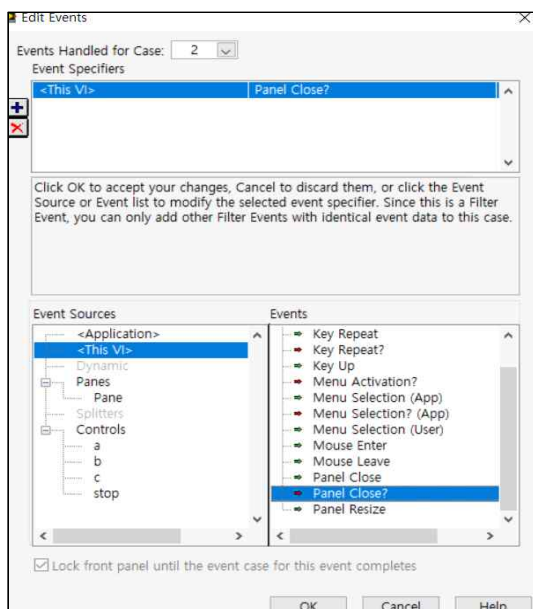
- **빨간색 화살표** : **필터 이벤트**로써, 발생 된 이벤트를 필터링 가능합니다. 필터 이벤트는 이벤트 데이터 노드와 이벤트 필터 노드를 모두 사용가능합니다.
- 예)프론트 패널의 라벨 ‘a’를 가진 컨트롤에 ‘마우스 다운’이 발생할 때, 특정 코드를 실행하고자 할 때, 이벤트 소스에서 ‘a’를 선택하고 이벤트에서 ‘마우스 다운’을 선택하면 이벤트지정자에 등록이 됩니다. 이처럼 이벤트 구조에서 **제공하는 이벤트 중 하나를 사용하는 것이 정적 이벤트**입니다.



- (가) : 타임아웃, (나) : 이벤트 데이터 노드, (다) : 이벤트 선택자 라벨 = 어떤 이벤트인지를 구별

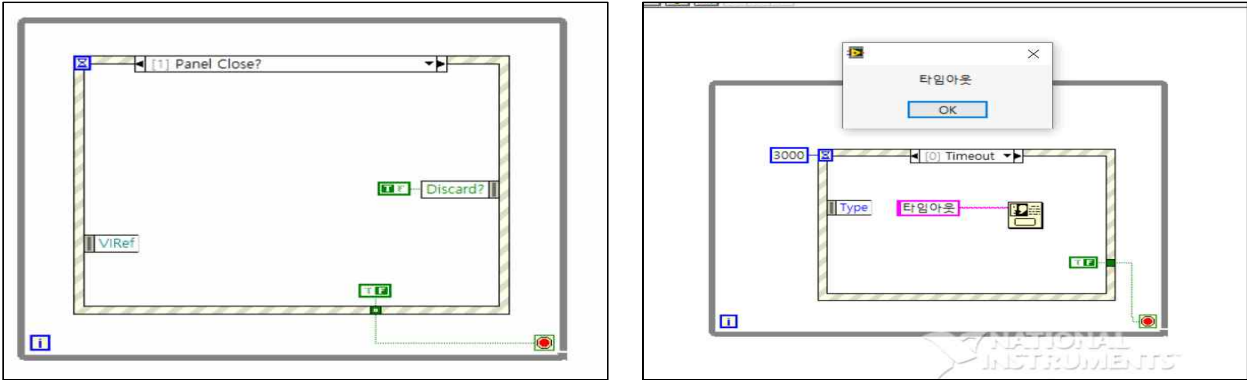


- 프로그램 실행 중 우측 상단에 'x'버튼을 누르면 프로그램이 닫히게 됩니다. 필터 이벤트로 'x'버튼을 눌러 닫히지 않고, 정상적으로 실행되도록 코드를 만들도록 하겠습니다.

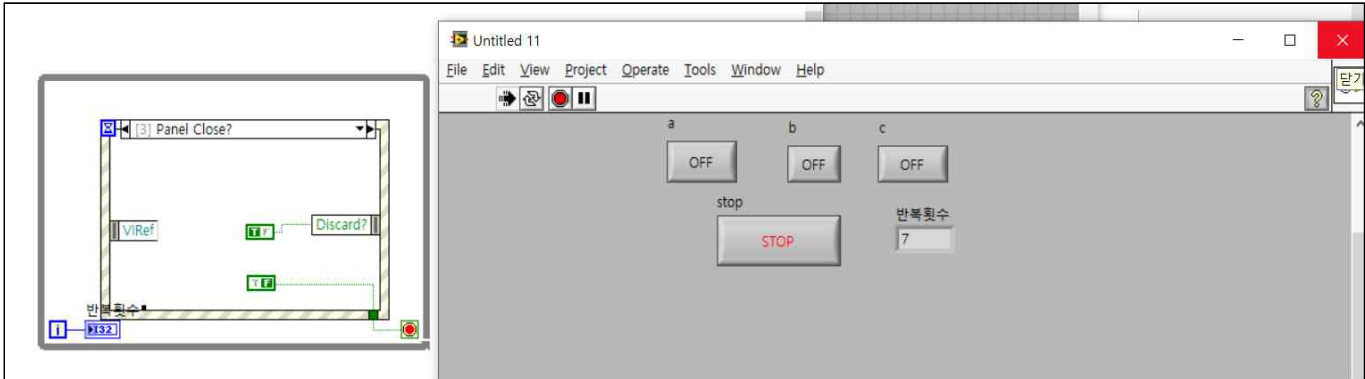


- 필터 이벤트를 선택하였기 때문에, 이벤트 구조에 이벤트 필터 노드가 나타나고 '버림?'에 참 상수를 연결해 주면

‘패널 닫기’버튼이 눌러진 이벤트를 버리게 됩니다. : 필터링하여 실행 중인 프로그램은 ‘x’버튼을 눌러도 닫히지 않고 정상적으로 실행하도록 하기 위하여 거짓 상수를 while 루프의 조건 터미널에 와이어링합니다.



- 오른쪽 : 타임아웃 이벤트 설정. 이벤트 구조의 좌측 상단에 있는 모래시계 모양에 ‘3000(msec)’의 값을 연결하였습니다. 아무 값도 주지 않으면 사용자가 정의해 놓은 이벤트가 발생하기를 무한 대기하는 반면에, 구체적인 시간 값을 주고 주어진 시간 내에 이벤트가 발생하지 않게 되면, 타임아웃 이벤트 케이스 안에 구현된 코드가 실행됩니다.
- 즉, 3000(msec)=3초 안에 정의한 이벤트가 발생 되지 않으면 ‘타임아웃’이라는 “메시지 창”이 뜰 것입니다.



- “VI”를 저장하고 실행하면 상단에 있는 ‘x’ 버튼을 눌러도 창이 닫히지 않는 것과 ‘c’ 버튼을 눌렀을 때, ‘b mouse up’이라는 대화 상자가 나타나는 것을 확인하고 프로그램을 정지합니다.

다이어그램 비활성화 구조 : 활성화와 비활성화로 나뉘져 있습니다. 이것은 주로 디버깅할 때 많이 사용됩니다. 아직 미완성의 코드를 일부만 실행시키고 나머지는 실행시키지 않을 때 쓰면 유용합니다.

- 비활성화 안에 구현된 노드들은 실행되지 않습니다.

