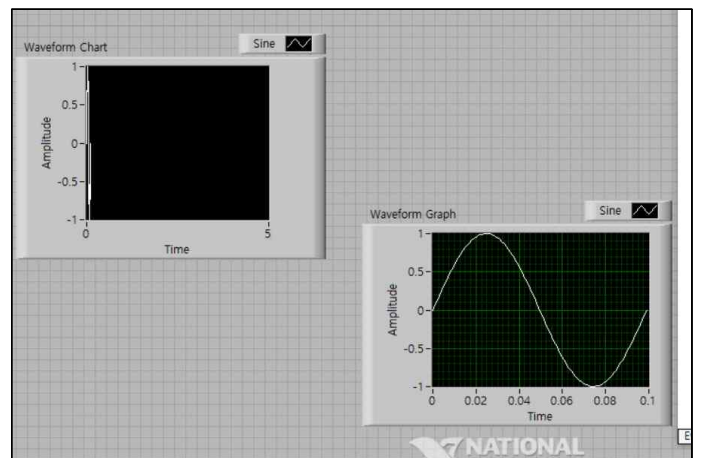
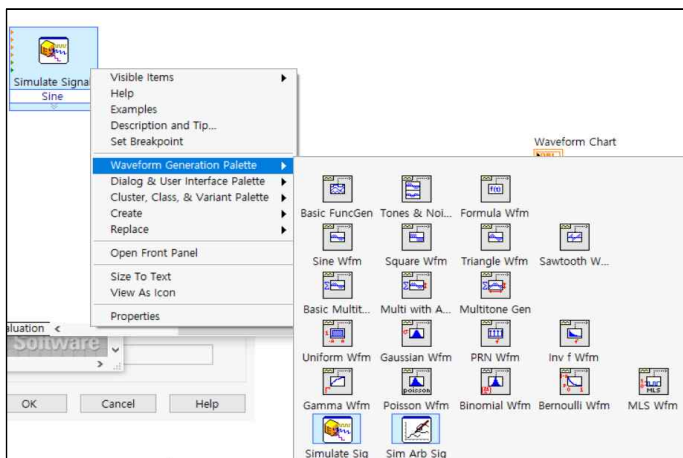
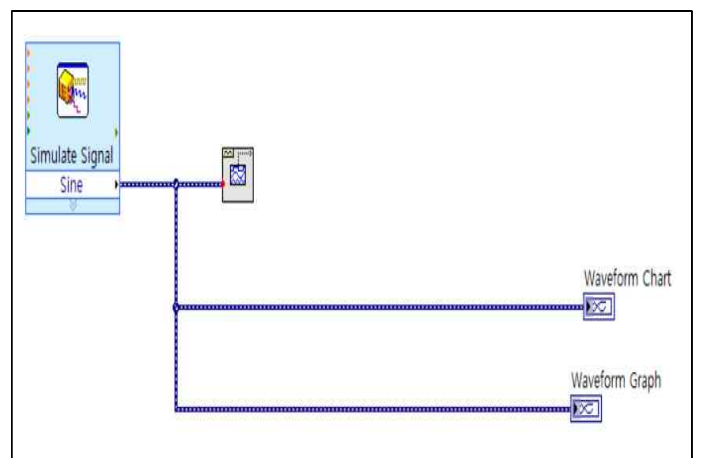


- 웨이브 폼 차트 : 과거 데이터가 남아 있어 누적된 데이터를 볼 수 있고, 웨이브 폼 그래프는 현재 데이터만 보입니다. (데이터 히스토리나 트렌드를 보기 위해서는 웨이브 폼 차트 [스칼라 연결]를 이용하고, 히스토리가 필요 없고 현재 데이터만 보려면 웨이브 폼 그래프 [배열 연결]를 사용합니다.)

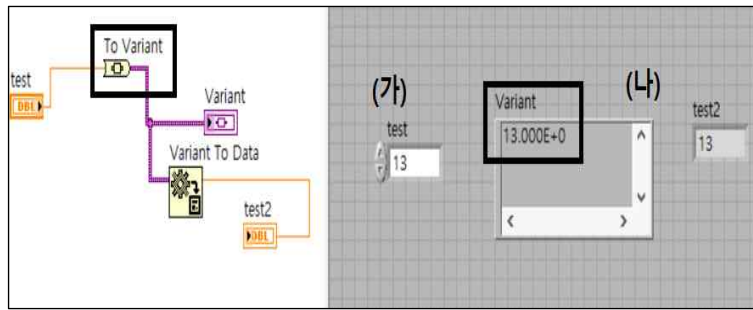
다이나믹 데이터 타입 : Dynamic data



- 익스프레스 노드들의 전용 데이터 타입, 터미널 색상이 짙은 파란색입니다.
- 웨이브 폼 데이터 : 데이터를 측정할 때, 측정 초기 시간, 측정 간격, 측정 데이터를 묶은 것을 의미합니다.



- 열거형 : 사용자가 정의한 데이터 중 **하나를 선택하고자 할 때** 사용하면 유용합니다.
- **variant** 데이터 타입 : **프로그램의 유연성 제공**. 배리언트 데이터를 사용하면, 프로그램을 유지 관리하기가 편리합니다.

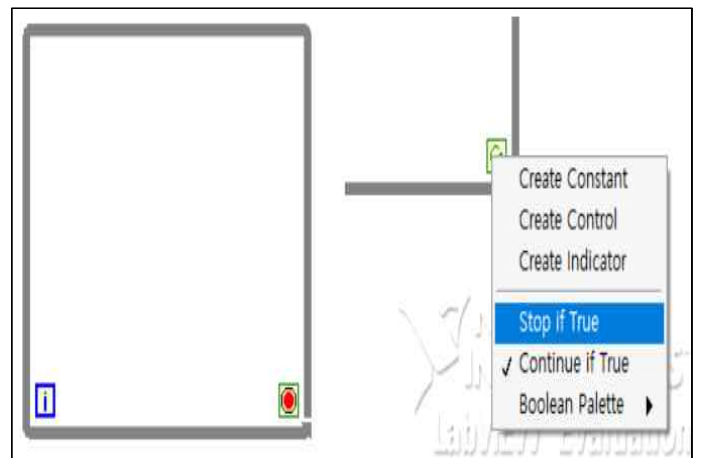
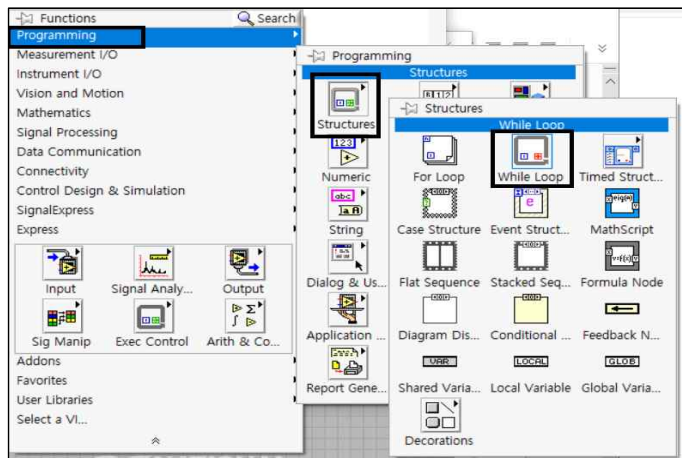


(가)에서 숫자형을 배리언트로 변환하고, (나)에서 배리언트를 숫자형으로 다시 원래의 데이터 타입으로 복원하는 예제.

- 타입 정의 : 사용자가 나만의 데이터 타입을 만드는 것. => 컨트롤 팔레트에서 제공하는 컨트롤/인디케이터를 쉽게 사용하듯이 사용자가 만든 데이터 타입을 저장해 두었다가 필요할 때 언제든지 사용할 수 있는 장점이 있습니다.
- 타입 정의 : 사용자가 정의한 데이터 타입을 언제 어디서든 쉽게 사용할 수 있는 장점이 있습니다.

4. 구조

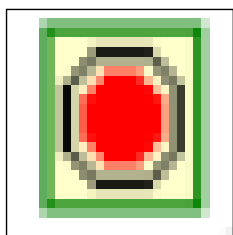
1) while 루프



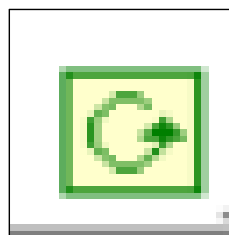
- 반복 실행을 위한 구조, While 루프 내부에 위치한 코드들은 정지 조건이 만족할 때까지 반복적으로 실행됩니다.

(Block Diagram - functions : Programming - Structures - "while loop")

- 반복 터미널 : 현재 몇 번 반복이 이뤄지고 있는지를 나타냅니다.
- 조건 터미널 : While 루프가 정지하는 조건을 정의. => 1. 참인 경우 정지, 2. 계속 두 가지 옵션이 있습니다.



참인 경우 정지 : Stop if True



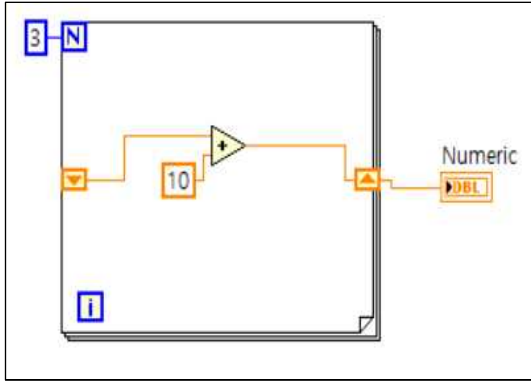
참인 경우 계속 : Continue if True

- while 루프는 코드를 먼저 실행하고 조건 터미널의 조건을 판단합니다.
- For 루프 : 몇 번 반복을 할 것인지를 지정해야 합니다.

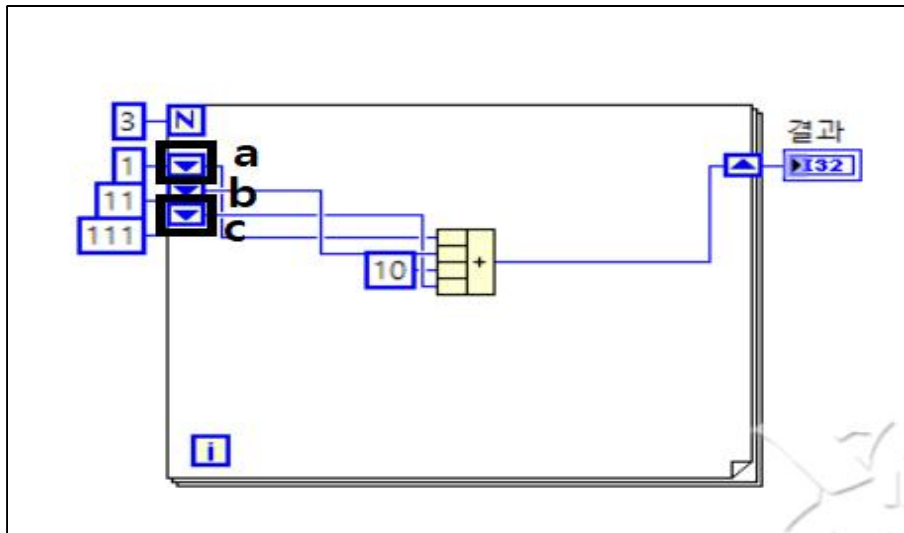


- 반복 구조(While/For)의 시프트 레지스터 : 시프트 레지스터는 반복 구문, 즉 While/For 루프에서 사용 가능.

시프트 레지스터 : 초기화 유무에 따라서 다른 결과값을 가집니다.



초기화 안 된 시프트 레지스터 : 초기화를 하지 않은 경우는 처음 실행하면 시프트 레지스터에 들어가는 값이 '0'입니다. 그리고 '출력'은 '30'이 나오고, 두 번째 실행하면 처음 실행했을 때 마지막 반복 즉 세 번째 반복을 해서 나온 '30' 값이 시프트 레지스터의 초기 값으로 사용되어 '출력'은 '60'이 됩니다.



- 시프트 레지스터의 왼쪽에 원소를 추가한 모습

- 원소를 추가하면, 두 번째 생긴 원소는 첫 번째 시프트 레지스터의 값을 받아 옵니다.

반복 횟수	a	b	c	결과
1	1	11	111	$1+11+111+10=133$
2	133	1	11	$133+1+11+10=155$
3	155	133	1	299

