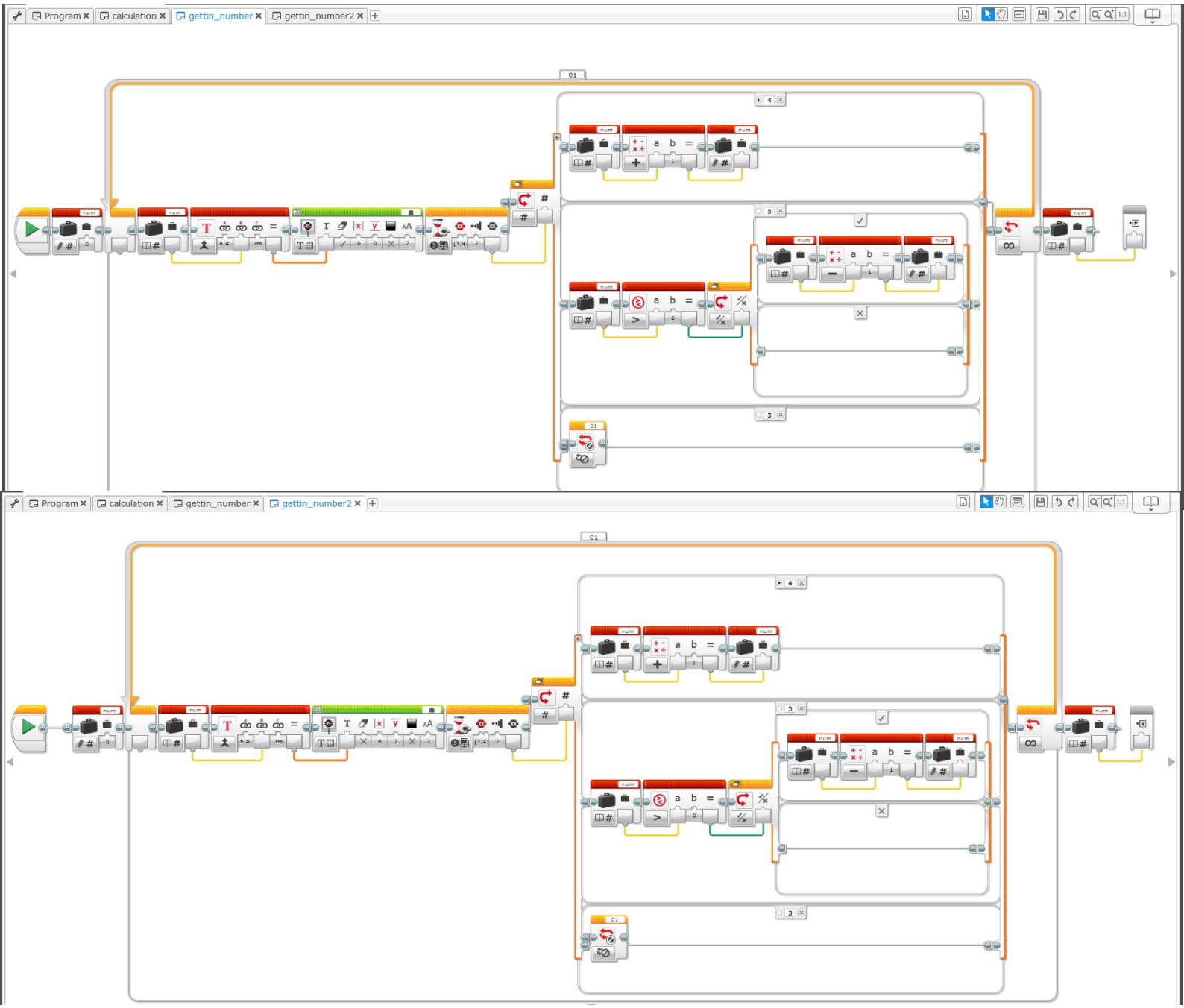
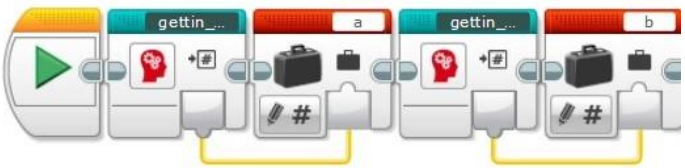
<마이블럭 getting_number, getting_number2>

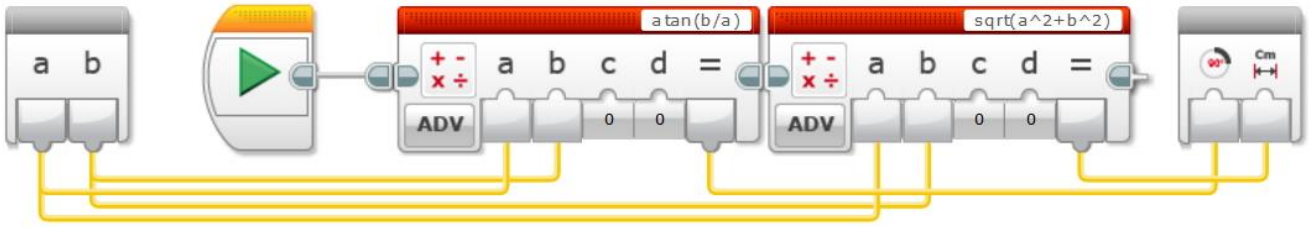


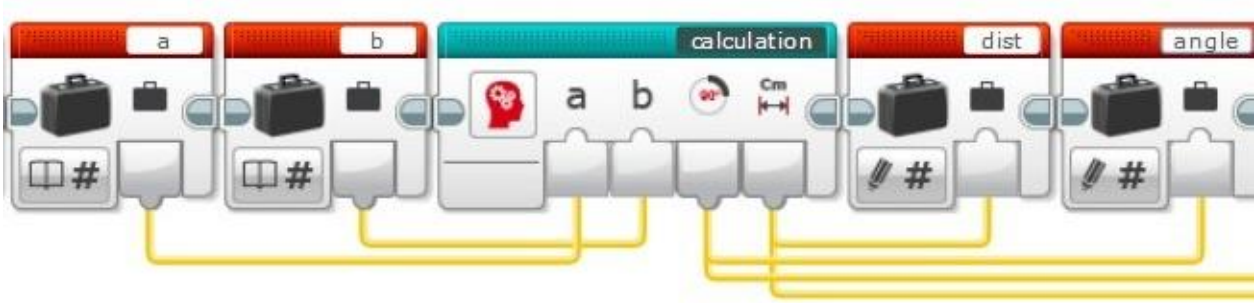


프로그램에서 숫자를 입력하는 것이 아닌 Ev3에서 직접 숫자 a, b를 입력 받도록 하기 위해서 이러한 프로그램을 구성하게 되었다. 직접 숫자를 입력 받는 프로그램이 꽤 길고 크기 때문에 전체 프로그램의 크기를 키우는 것을 방지하고 동일한 작업을 반복하는 것을 막기 위해서 마이블록(gettin_number)으로 제작하게 되었다. a, b를 각각 입력 받는 마이블록을 설정하였다. 구성은 동일하나 화면에 표시하는 내용이 다르기 때문에 이와 같이 설정하였다. 두개의 마이블록의 차이점은 화면에 표시할 때 a냐 b냐의 차이와 텍스트의 줄바꿈정도의 차이가 있다. 두개의 마이블럭은 a, b 값을 설정하는 것으로 먼저 앞에서 변수 num=0이라는 값을 초기화 하는 것으로 시작한다. 이는 앞에서 a값이 설정되고 b값을 설정할 때 b값의 초기값을 0으로 해주기 위함이다. 이후 num이라는 변수를

a= num과 같은 형식으로 병합시켜서 실시간으로 현재의 num값을 디스플레이에 출력한다. 그 다음 루프는 브릭 버튼이 눌릴 때까지 대기시간을 갖게 된다. 브릭버튼이 눌리면 그 버튼의 id가 생성되게 되는데 그 id에 따라서 스위치블록이 나누어지게 된다. 위 버튼의 id인 4번으로 가면 num에 1이 더해지게 되고, 아래 버튼인 5번은 1을 빼게 되고, 숫자를 다 입력했고 다음 숫자를 정하겠다는 표지인 오른쪽 버튼은 3번으로 루프 인터럽트를 하게 된다. 단, 아래 버튼을 눌러서 1을 빼게 될 때 num을 양수로 유지하기 위해서 num이 0보다 클 때만 1을 빼도록 설계하였다 여기서 받은 num은 출력 번수로 빠져나가며 나간 후 a, b에 순차적으로 저장되게 된다.

<마이블럭 calculation>

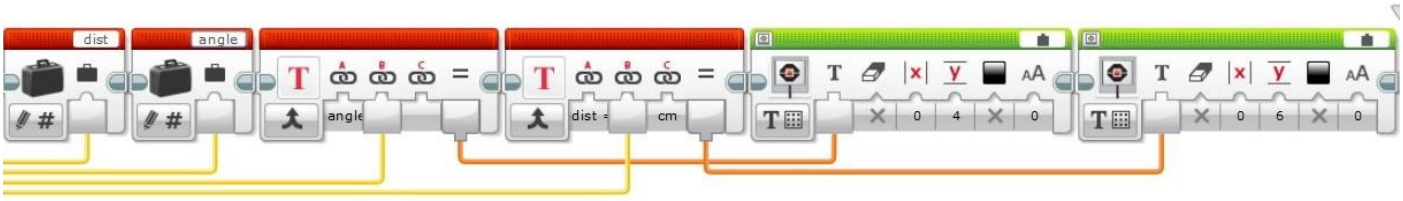




마이블럭을 통해 입력 받은 수는 순차적으로 변수 a, b에 저장되게 되고 이는 마이블럭 calculation으로 들어가게 된다. calculation에서는 입력 받은 a, b를 가지고 연산을 해서 돌아야 하는 각도와 거리를 계산한다. 각도는 탄젠트(b/a)의 역함수로 구할 수 있으며 이 값이 양수로만 나오게 하기 위해서 위의 마이블럭 getting_number에서 a, b를 양수로만 받게 설정하였다. 거리는 피타고라스 공식인 제곱근(a제곱+b제곱) 을 사용하여 구하였다. 결과값들은 각각 angle변수와 dist 출력 변수로 들어가게 되고 이후 마이블럭을 나와 angle변수와 dist 변수에 순차적으로 저장되게 된다.

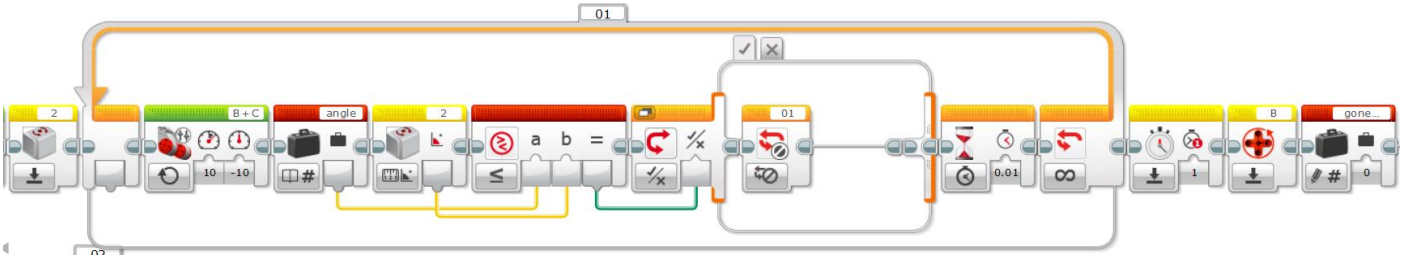
<전체 프로그램>

<디스플레이 출력(계산된 거리, 각도)>



angle, dist변수에 각각 각도, 가야할 거리가 표시된 이후에 angle = angle값, dist = dist값 cm 가 각각 병합되어 계산된 값을 디스플레이에 출력해 계산적으로 돌아야 할 각도와 가야할 거리를 나타낸다.

<회전>



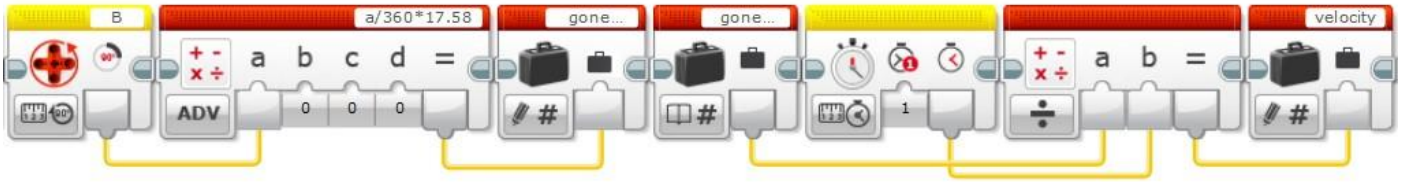
회전 루프<01>에서 브릭은 오른쪽으로 포인트 턴을 하게 되고 돌면서 대기시간을 넣어 줌으로서 스태핑 모터를 구현해 회전의 정확도를 좀 더 높였다. 자이로센서의 정확성을 위해서 루프 앞에서 자이로센서의 값을 초기화 시켰다. 초기화 시킨 후 자이로센서는 측정을 시작하며 비교 연산에서 angle의 저장된 값과 자이로센서에 측정된 회전 값을 비교한다. 만약 회전한 값이 angle값과 같거나 보다 커질 경우 스위치블록의 참 값이 실행 되며 루프를 탈출한다.

<회전 후 간 거리, 속도 측정 정확성 향상>

측정을 보다 정확히 하기 위해 타이머01의 값, 바퀴b의 회전 값, 그리고 간 거리 변수 값 gone distance를 모두 초기화 시켰다.

<주행>

<거리, 속력 측정>



지나간 거리의 계산은 바퀴가 회전한 각도를 기반으로 하였다. 센서를 통해 측정된 바퀴의 회전한 각도는 라디안이 아닌 도로 계산 되며 이 각도를 360도로 나누게 되면 바퀴가 회전한 회전수가 나오게 된다. 이 회전 수를 바퀴의 둘레의 길이인 17.58에 곱해주게 되면 지나간 거리가 계산된다. 이렇게 계산된 거리는 gone_distance변수에 저장된다.

속력의 계산은 타이머에 저장된 시간과 앞의 변수 gone_distance를 이용한다. 속력은 거리/시간이므로 이를 이용해서 계산을 하고 결과 값을 변수 velocity에 저장한다.

<거리, 속력 표시>

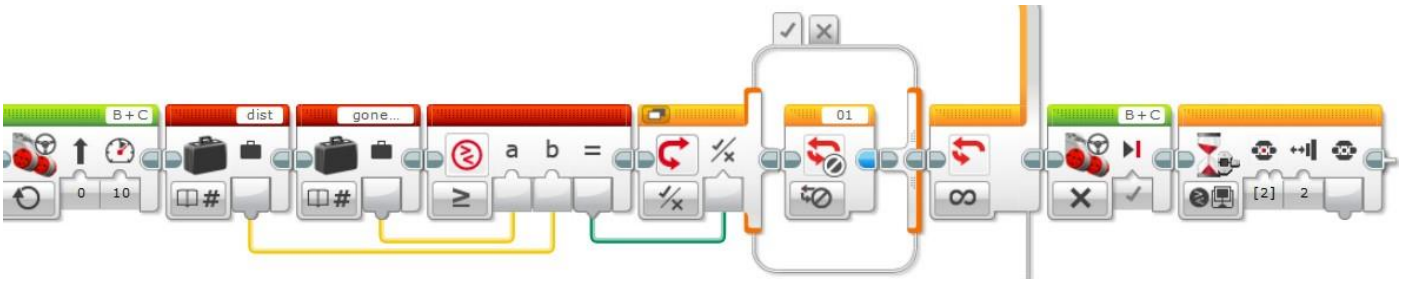
지나간 거리는 gone_distance에 저장된 변수를 텍스트 병합을 이용해 walked gone_distance값 cm 의 형식

으로 만들어서 디스플레이에 출력한다.

속력도 마찬가지로 velocity에 저장된 변수를 텍스트 병합을 이용해 v = velocity값 cm/s 의 형식으로 만들어서 디스플레이에 출력한다.

모든 디스플레이의 출력은 지우기 기능을 제거해 모든 정보가 창에 뜨도록 했다.

<직진>

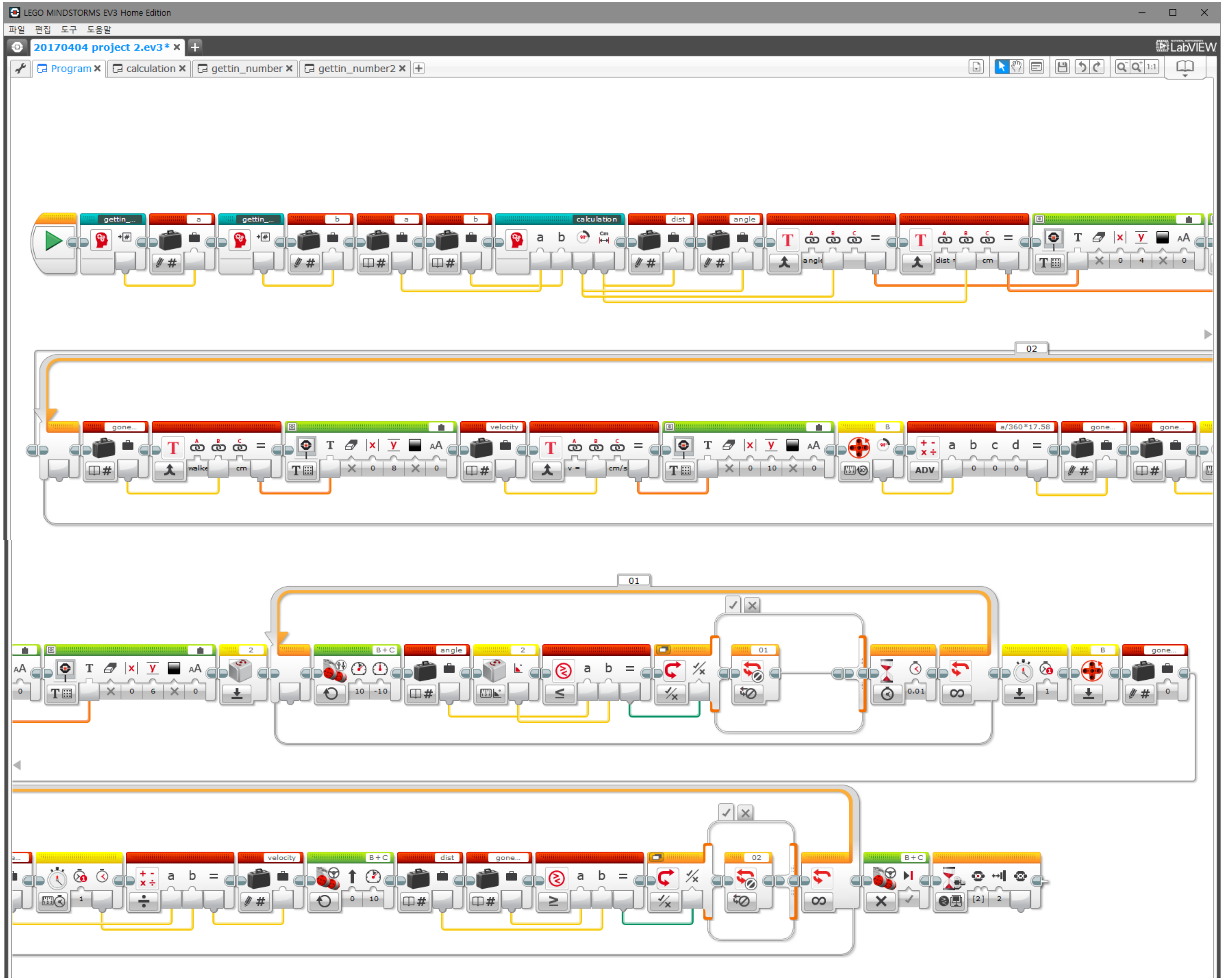


라지모터를 주행 모드로 속도를 10, 방향을 앞으로 설정했다. 이후 만약 gone_distance가 dist값과 같거나 넘어서게 되면 비교 값이 참이 되게 되고 이는 곧 스위치 블록의 참값이 실행되게끔 한다. 스위치 블록에는 루프 인터럽트가 있어 루프를 빠져 나오게 되고, 루프 밖에는 모터 정지와 브릭 버튼이 눌릴 때까지 대기하는 블록이 있어, 주어진 거리를 모두 주행한 이후에 정지하고 확인으로 브릭 가운데 버튼이 눌릴때까지 디스플레이를 출력하며 정지한다.

〈추가 기능 구현에 대한 제안내용〉

좌표 계에서 어떤 점이 찍히면 그 점에 대해 그 점으로 가는 최단거리를 구할 수 있고 그 점과 x축과 이루는 각도, 그 점으로 최소한의 시간으로 가장 빠르게 이동하는 경로를 알아낼 수 있다. 또한 그 경로로 로봇을 주행시킬 수 있다.

이는 비행기 항로의 설정, 공장에서의 컨베이어 벨트, 기타 거리의 효율성이 요구되는 상황에 응용 될 수 있다.


LEGO MINDSTORMS EV3 Home Edition
파일 편집 도구 도움말
20170404 project 2.ev3*
Program
calculation
gettin_number
gettin_number2
LabVIEW
gettin_...
a
b
calculation
dist
angle
velocity
gone...
a/360*17.58
B+C
02
01