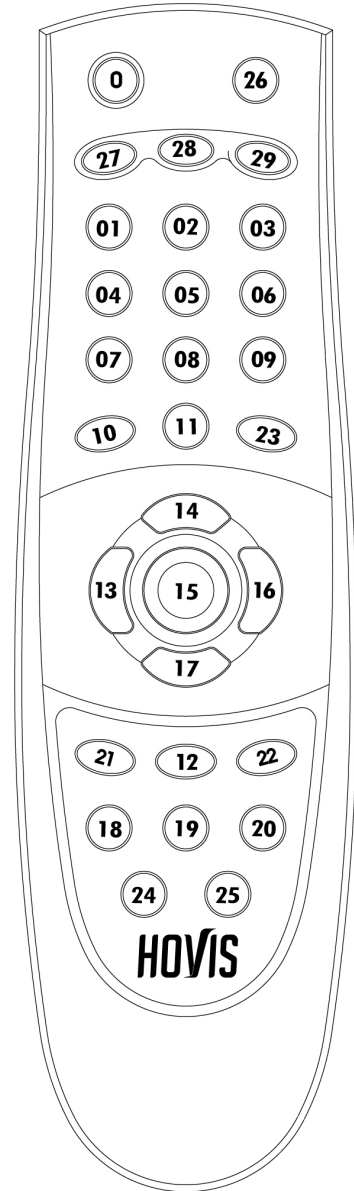
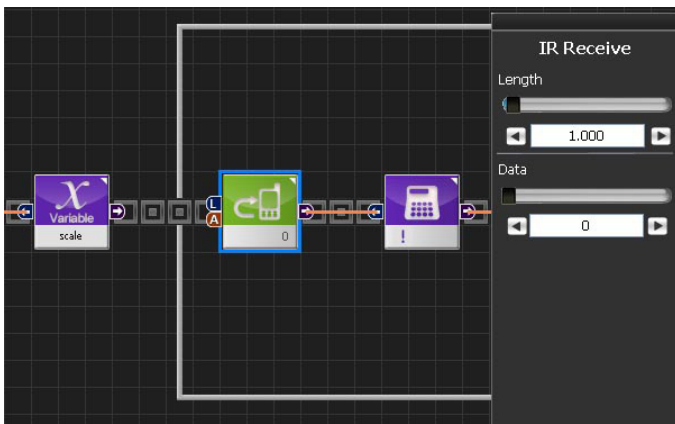


IRReceive, Sound & Motion 예제 따라하기

(Sound 예제로 설명, Motion 예제 설명생략 리모콘 번호 매칭)

IRReceiv Module 의 Data 값의 번호는 오른쪽 리모콘의 키와 매칭됩니다.

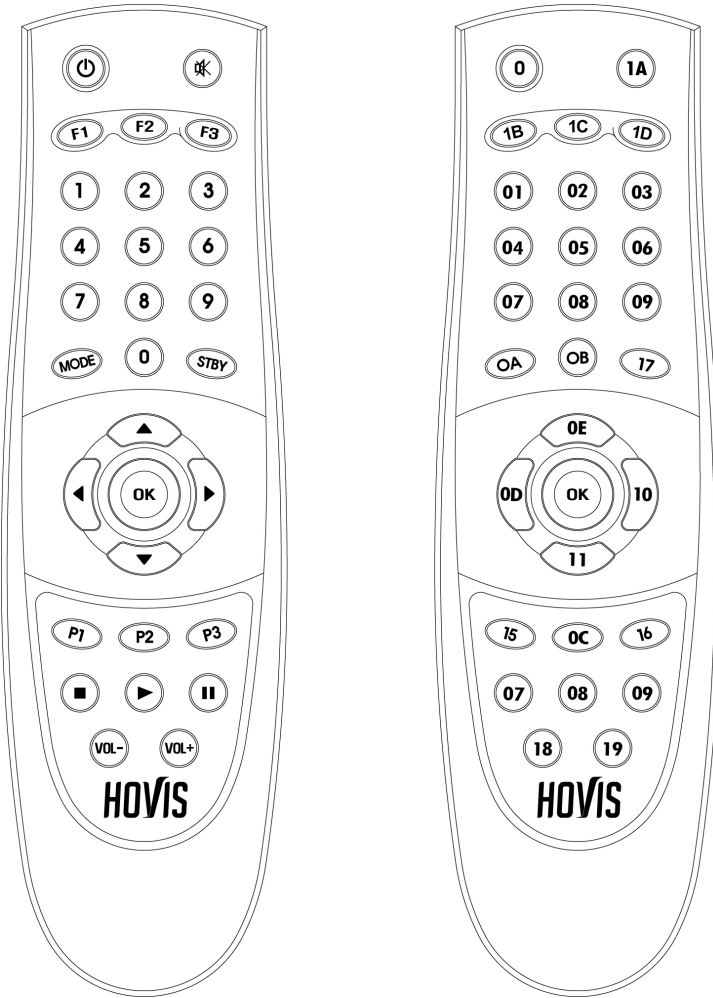


Hovis 리모컨의 키맵은 왼쪽 그림과 같습니다. IR Receive 모듈의 Data가 왼쪽의 키 숫자에 대응됩니다.

가령 왼쪽 위의 전원 버튼을 누르면 Data 0이 DRC로 들어오게 됩니다. 이 때 사용자는 IR Receive 모듈에서 Data를 0으로 설정해서 Switch 모듈의 입력으로 연결하면, 전원 버튼을 눌렀을 때 로봇이 어떤 행동을 할 것인지를 프로그래밍 할 수 있습니다.

리모컨에는 채널이 있어서, 리모컨의 채널과 DRC에 설정 된 채널이 같아야만 리모컨 데이터를 받을 수 있습니다.

리모컨의 채널은 1~0의 숫자 + OK를 동시에 누름으로써 설정할 수 있으며, DRC의 채널은 MPSU Ram Data의 RmcChannel 값을 바꾸어 설정할 수 있습니다. 리모컨의 숫자에 대응되는 RmcChannel 값은 아래와 같습니다.

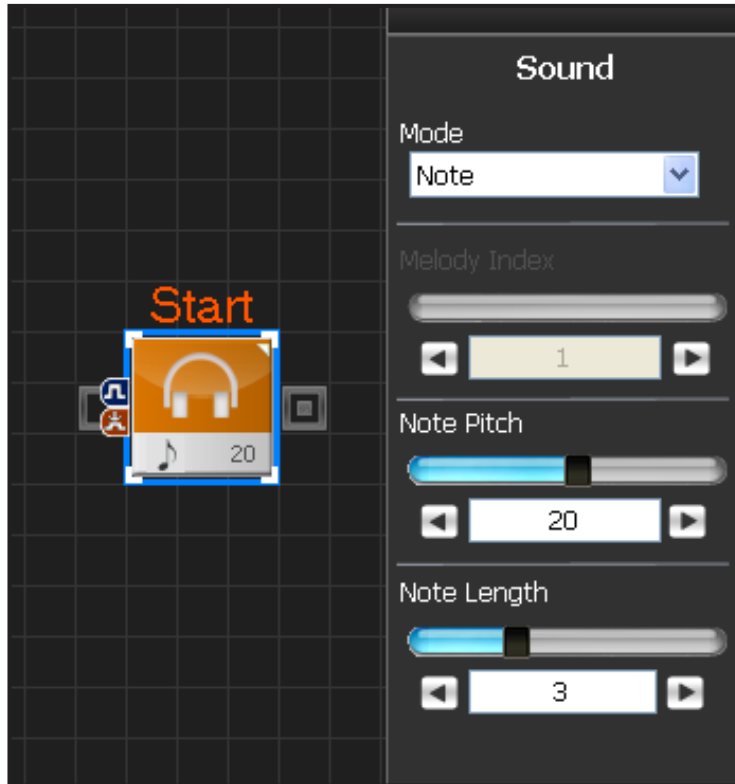


리모컨 버튼	RmcChannel 값
0+OK	97(0x61)
1+OK	98(0x62)
2+OK	99(0x63)
3+OK	100(0x64)
4+OK	101(0x65)
5+OK	102(0x66)
6+OK	103(0x67)
7+OK	104(0x68)
8+OK	105(0x69)
9+OK	106(0x6A)

예제설명

리모컨의 번호와 Sound 음을 매칭시켜서 도레미파솔라시도 (1번~8번) 음이 나오게 하는 프로그램입니다.

도레미 음은 Motion > Sound 모듈에서 Note의 Pitch 값에 따라 음이 달라집니다. DRC 제거기 음높이는 총 0번에서 37번까지 있으며, 총 3옥타브까지 음을 낼 수 있습니다.



00 Sound 속성창

Motion > Sound 모듈을 선택합니다.

Mode 에는 Melody 와 Note 가 있습니다. Melody 는 저장되어있는 편집음 리스트중에 하나를 선택하여 플레이합니다.

Mode 에서 Note 는 자체적인 36개의 음을 가져와서 사용할 때 선택합니다.

아래 표를 참조하세요

Note Pitch 는 0번에서 37번까지 선택합니다. 총 3옥타브로 구성됩니다.

Note Length 는 음의 박자를 의미합니다. 32분음표에서 온음표까지 선택할 수 있습니다.

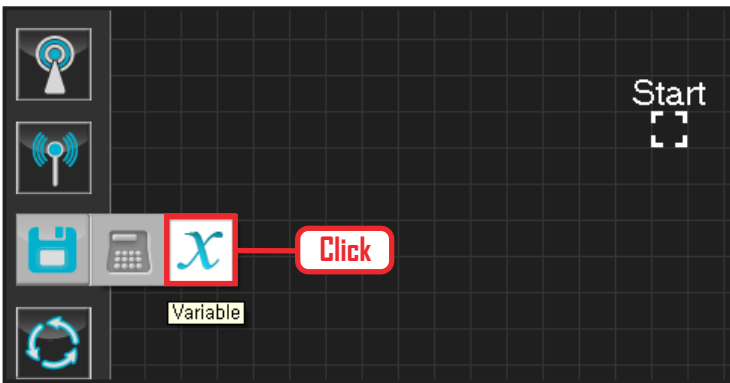
아래 표를 참조하세요.

Note Pitch (음높이)

번호	0												
계명	무음												
번호	1	2	3	4	5	6	7	8	9	10	11	12	
계명	도	도#	레	레#	미	파	파#	솔	솔#	라	라#	시	
번호	13	14	15	16	17	18	19	20	21	22	23	24	
계명	도	도#	레	레#	미	파	파#	솔	솔#	라	라#	시	
번호	25	26	27	28	29	30	31	32	33	34	35	36	37
계명	도	도#	레	레#	미	파	파#	솔	솔#	라	라#	시	도

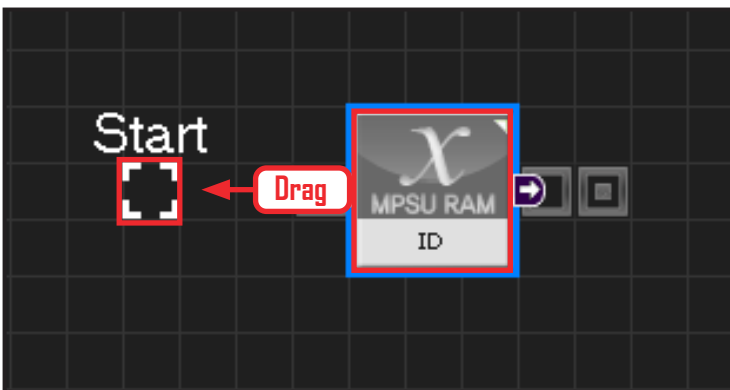
Note Length (음길이)

번호	0	1	2	3	4	5	6	7	8	9
Raw Data	6	12	18	24	36	48	72	96	144	192
시간 (ms)	38.4	76.8	115.2	153.6	230.4	307.2	460.8	614.4	921.6	1228.8
음표	32분 음표	16분 음표	점16분 음표	8분 음표	점8분 음표	4분 음표	점4분 음표	2분 음표	점2분 음표	온음표



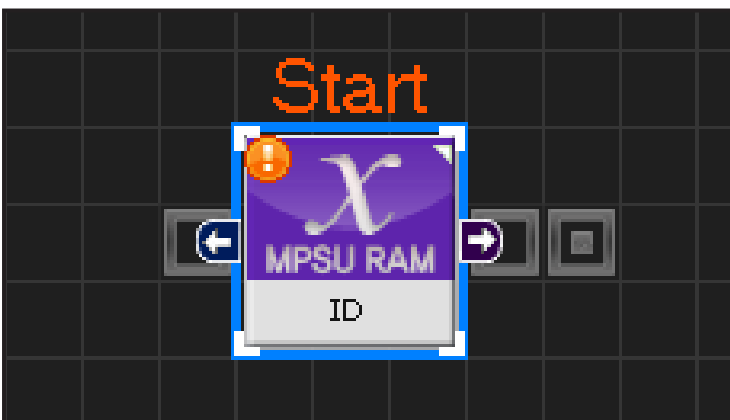
01 변수 지정

Data > Variable 모듈을 클릭합니다.



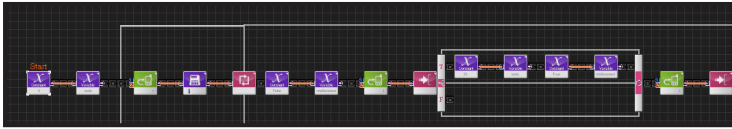
02 시작

모듈의 왼쪽 연결선을 Start Point 에 드래그하여 정확히 도킹을 시킵니다.



03 프로그래밍 시작

모듈과 Start Point 가 정확히 도킹하면 왼쪽과 같이 활성화된 칼라 이미지 모듈로 변합니다. 그럼 프로그래밍이 시작되었다는 의미입니다.



04 전체 프로그래밍

리모컨과 버저를 이용한 전체 프로그래밍입니다.

◀ ▶
C-like
Graphic

```

1 void main()
2 {
3     scale=0
4     while( !( (MPSU_RmcLength >= 8 && MPSU_RmcData == 0) ) )
5     {
6         rmcReceived=false
7         if( (MPSU_RmcLength >= 0 && MPSU_RmcData == 1) )
8         {
9             scale=25
10            rmcReceived=true
11        }
12        else
13        {
14        }
15        if( (MPSU_RmcLength >= 0 && MPSU_RmcData == 2) )
16        {
17            scale=27
18            rmcReceived=true
19        }
20        else
21        {
22        }
23        if( (MPSU_RmcLength >= 0 && MPSU_RmcData == 3) )
24        {
25            scale=29
26            rmcReceived=true
27        }
28        else
29        {
30        }
31        if( (MPSU_RmcLength >= 0 && MPSU_RmcData == 4) )
32        {
33            scale=30
34            rmcReceived=true
35        }
36        else
            
```

05 C-Like 보기

오른쪽 상단의 Graphic 탭에서 C-like 탭을 클릭하면 왼쪽과 같은 Task 프로그래밍 화면이 나옵니다. IRReceive 리모컨과 사운드를 이용한 전체 프로그래밍 화면입니다.

C와 유사한 문법 구조를 가지고 있으므로 C 문법 선행학습 효과도 있습니다.

각 모듈별로 클릭하면 커서가 따라서 움직이므로 모듈별로 Text로 어떻게 변환하는지 확인할 수 있습니다

```

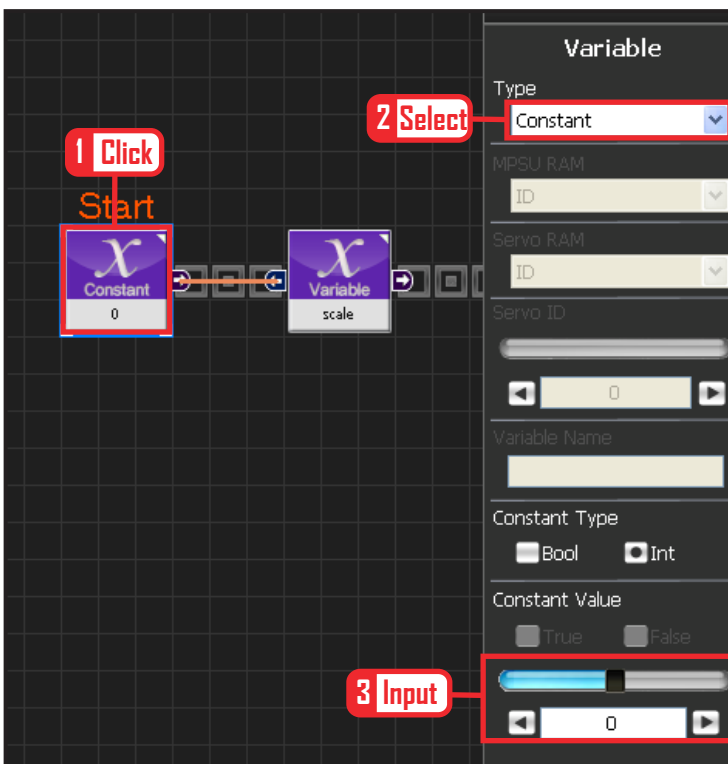
37 {
38 }
39 if( ( MPSU_RmcLength >= 0 && MPSU_RmcData == 5 ) )
40 {
41     scale=32
42     rmcReceived=true
43 }
44 else
45 {
46 }
47 if( ( MPSU_RmcLength >= 0 && MPSU_RmcData == 6 ) )
48 {
49     scale=34
50     rmcReceived=true
51 }
52 else
53 {
54 }
55 if( ( MPSU_RmcLength >= 0 && MPSU_RmcData == 7 ) )
56 {
57     scale=36
58     rmcReceived=true
59 }
60 else
61 {
62 }
63 if( ( MPSU_RmcLength >= 0 && MPSU_RmcData == 8 ) )
64 {
65     scale=37
66     rmcReceived=true
67 }
68 else

```

```

69 {
70 }
71 if( ( true == rmcReceived ) )
72 {
73     note( scale, 3 )
74     waitwhile( MPSU_BuzzTime )
75 }
76 else
77 {
78 }
79 }
80 }

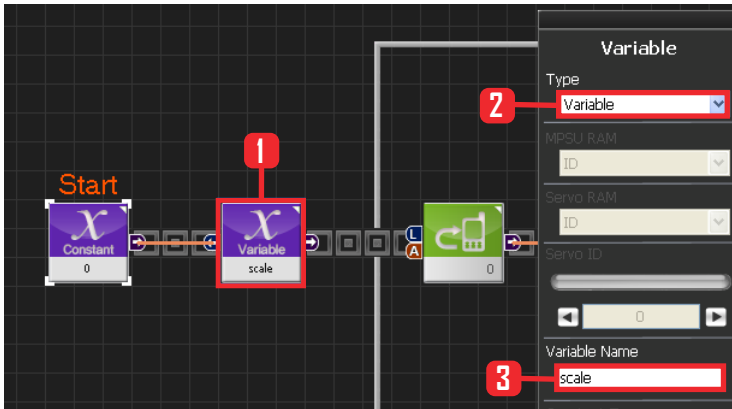
```



06 상수 설정

플레이할 음계의 변수를 선언하는 과정입니다.

Data)Variable 을 선택합니다.
 Type : Constant 로 선택합니다.
 Constant Value : 0 으로 설정합니다.



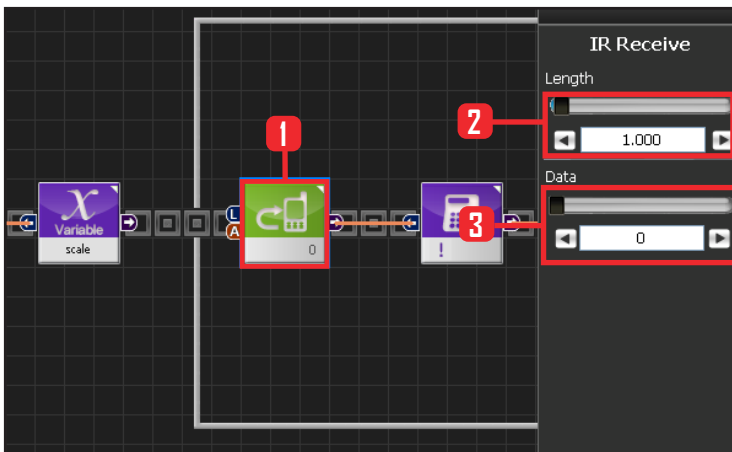
07 변수명

Play할 음계의 변수 이름을 지정합니다.

Data > Variable 을 선택합니다.

Type : Variable 로 선택합니다.

Variable Name : scale 로 입력합니다.



08 While 문 예외 조건

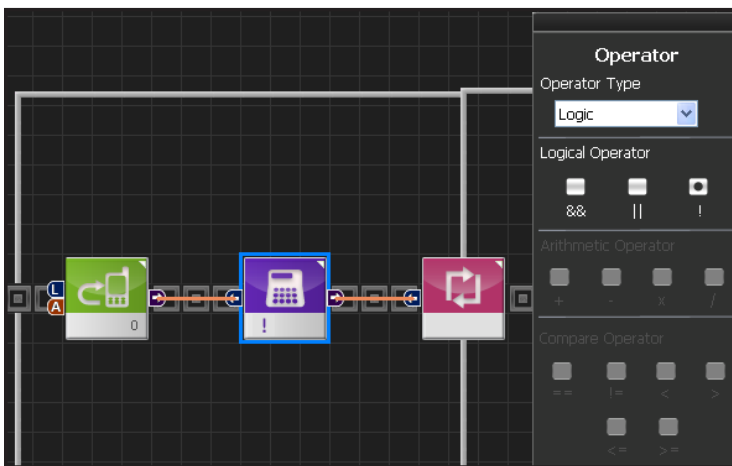
리모컨 0번을 길게 누르면 빠져나옵니다.

Communication)IRReceive 모듈을 선택합니다.

Length : 1.000 으로 설정합니다. 1초동안 누른다는 의미입니다.

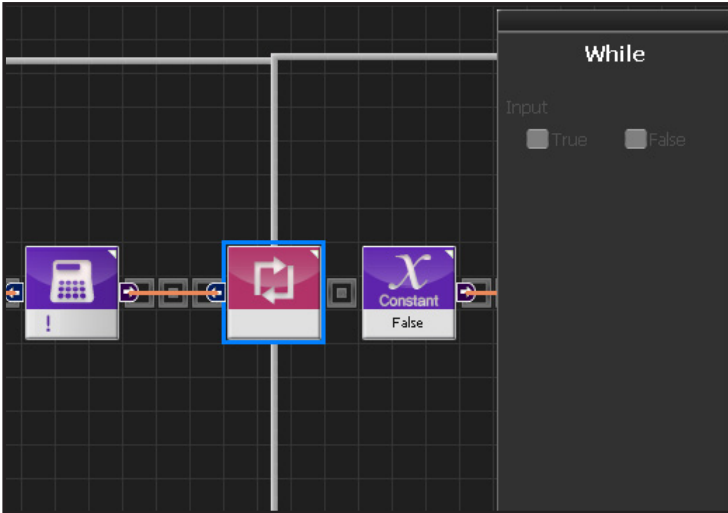
Data : 0 으로 설정합니다. 전원버튼을 누른다는 의미입니다.

전원버튼을 1초 이상 길게 누르면 이 모듈의 Output 값은 true가 되고, 아닌경우 false 입니다.



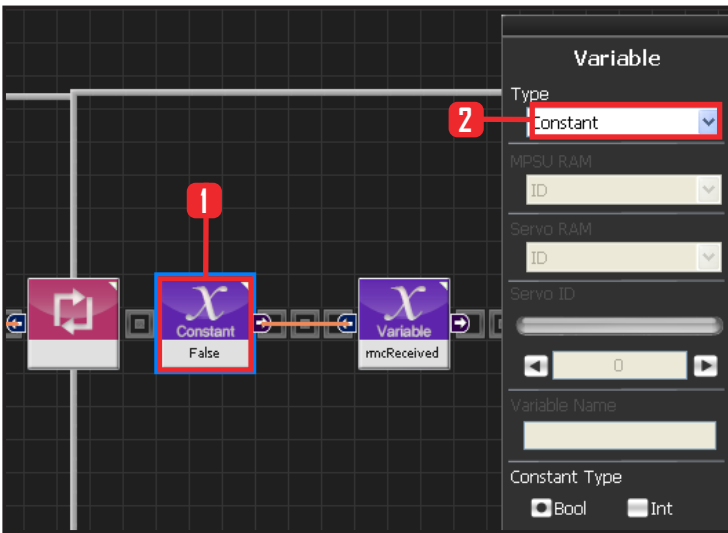
09 ! 연산자 설정

! 은 true / false 값을 반대로 만들어 줍니다. 따라서 IRReceive 모듈의 출력값이 반대가 되어 while문의 입력으로 들어갑니다.



10 While 조건반복

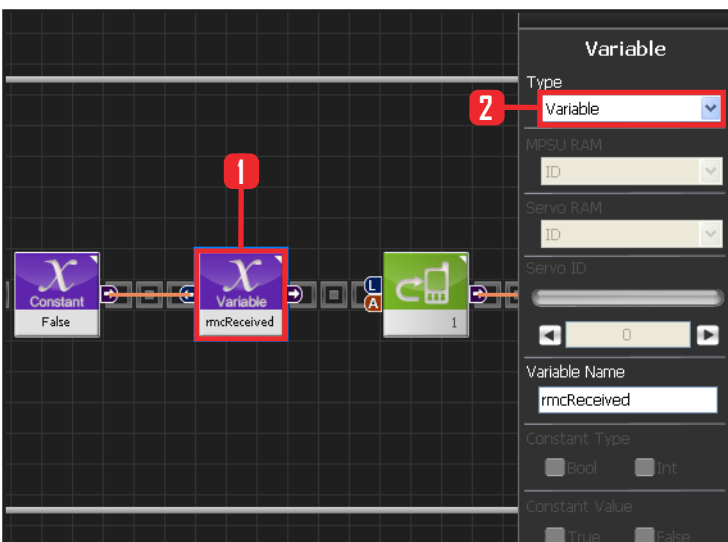
앞에 조건에 따라 반복을 합니다.
 입력이 true이면 계속 뒤의 내용을 반복합니다. !연산자를 거치므로 IRReceive 모듈의 Output이 false이면 반복하고, true 이면 loop를 빠져 나옵니다. 즉 전원 버튼을 1초 이상 누르면 반복문을 빠져나오는 것 입니다.



11 리모컨 입력 변수 초기화

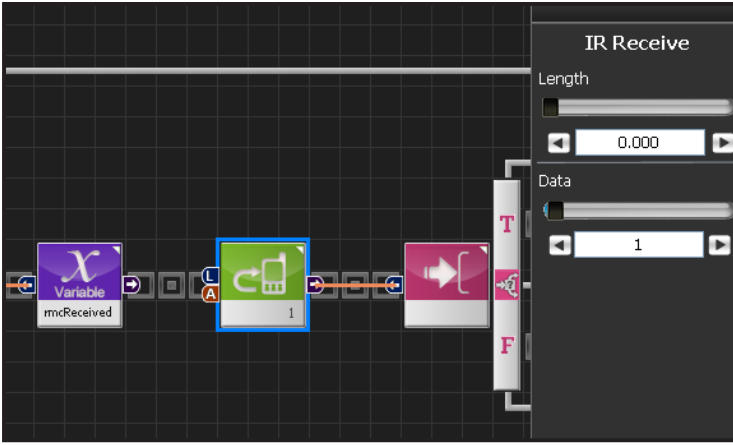
리모컨을 입력받았다는 변수를 설정합니다.

Data > Variable 모듈을 선택합니다.
 Type : Contant 를 선택합니다.
 Constant Type 은 Bool 로 설정합니다. Bool 은 참과 거짓을 나타내는 자료형입니다.
 Constant Value : False 를 선택합니다



12 리모컨 입력 초기 변수

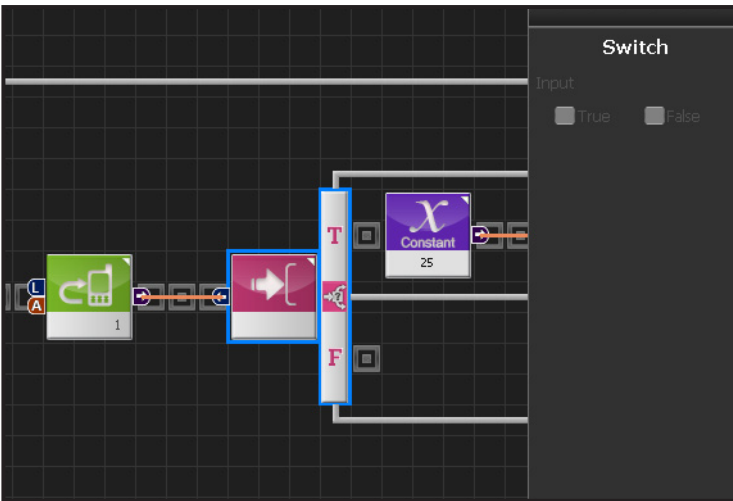
Data > Variable 을 선택합니다.
 Type : Variable 로 선택합니다.
 Variable Name : rmcReceived 로 입력합니다
 rmcReceived 는 loop에서 1~8의 리모컨 입력을 받았다는 것을 표시하는 변수명으로서, loop(반복) 처음에는 false로 초기화하고, loop(반복)이 끝날때쯤 검사해서 true 이면 음표가 플레이 됩니다.



13 1번 리모컨

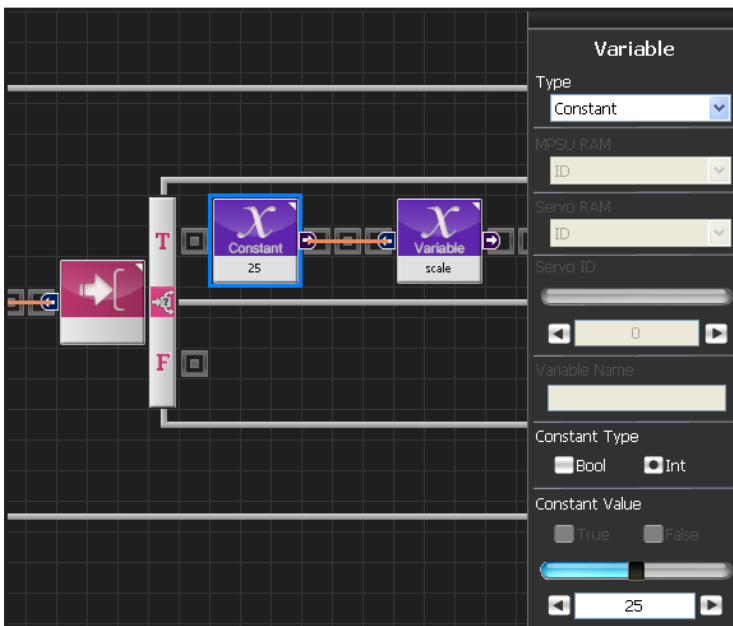
1번 리모컨이 눌렀는지 검사합니다.

Communication>IRReceive 모듈을 선택합니다.
Length : 0.000 으로 설정합니다.
Data : 1 로 설정합니다. 1번 버튼을 의미합니다..



14 IF 조건문

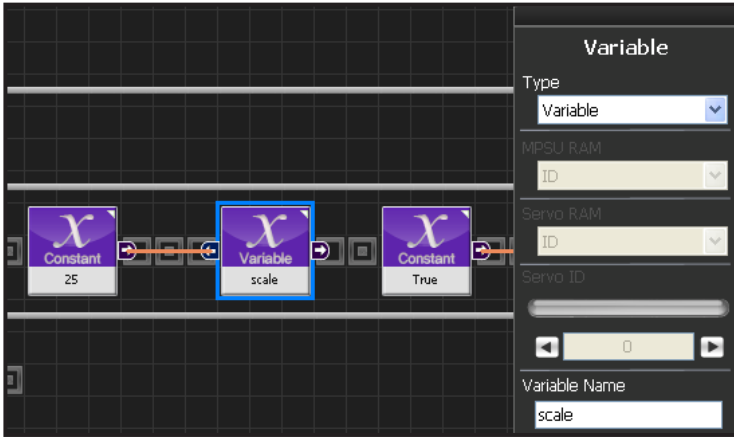
True 이면 실행합니다.



15 “도” 음 저장

앞에서 설명한 Note Pitch (3옥타브 음) 번호가 25 번이면 도 음을 의미합니다.
Scale 값을 도 로 바꿔줍니다.

Data>Variable 모듈을 선택합니다.
Type : Contant 를 선택합니다.
Constant Type 은 int 로 설정합니다.
Constant Value : 25 를 선택합니다. 25는 음계 “도” 를 나타냅니다.

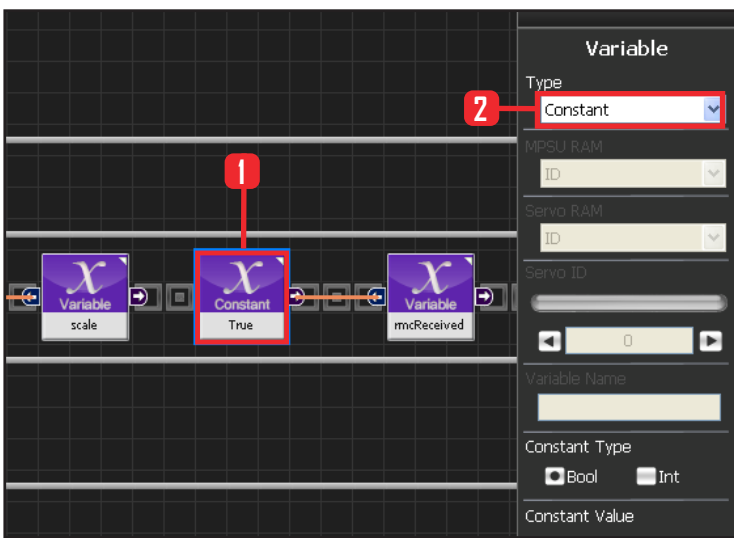


16 scale

Play할 음계의 변수 이름은 scale 로 동일합니다.

Data > Variable 을 선택합니다.
Type : Variable 로 선택합니다.
Variable Name : scale 로 입력합니다.

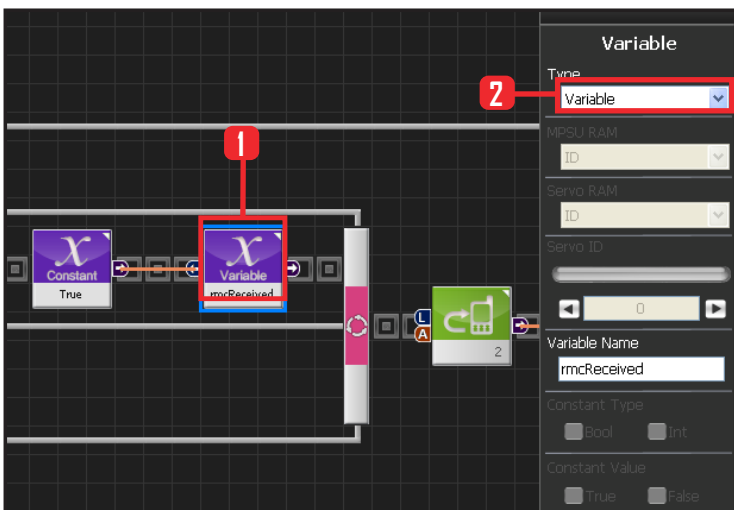
앞 상수 25 값을 input connector 로 입력받습니다.



17 리모컨 받았음을 저장

변수 rmcReieved 가 True 이면 1~8번 버튼중 하나가 눌렀다는 것을 의미합니다.

Data > Variable 모듈을 선택합니다.
Type : Contant 를 선택합니다.
Constant Type 은 Bool 로 설정합니다.
Constant Value : True 를 선택합니다.



18 리모컨 받았음을 저장

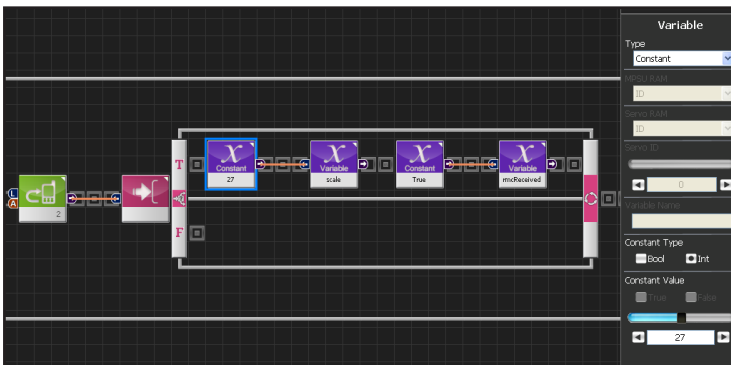
Data > Variable 을 선택합니다.
Type : Variable 로 선택합니다.
Variable Name : rmcReceived로 동일합니다.

앞상수 true값을 input connertor로 입력 받습니다.



19 1번 -> “도” 음

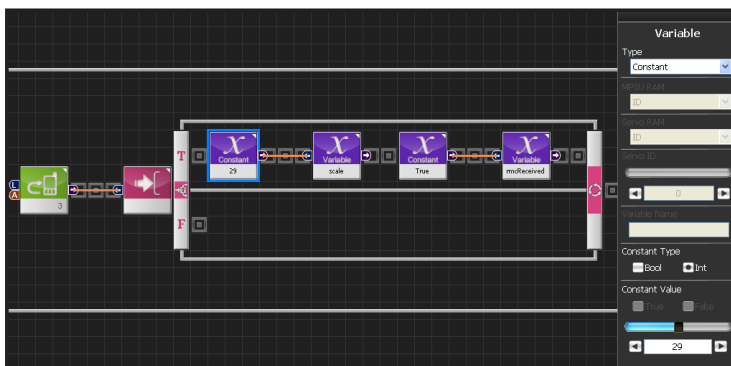
리모컨 1번을 눌렀을 때 ‘도’ 음을 Scale에 저장하는 프로그래밍 입니다.



20 2번 -> “레” 음

리모컨 2번을 눌렀을 때 ‘레’ 음을 Scale에 저장하는 프로그래밍 입니다.

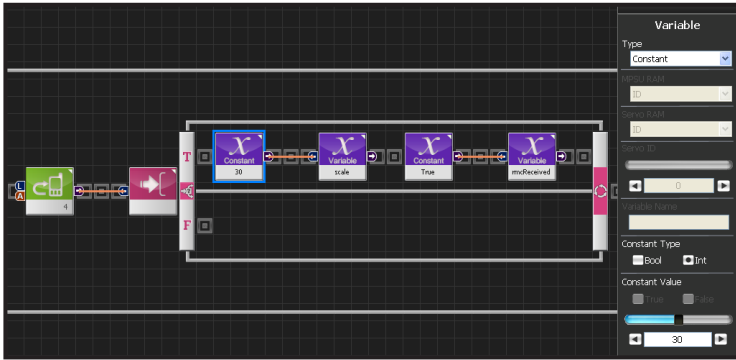
Scale = 27 번이 ‘레’ 음입니다.



21 3번 -> “미” 음

리모컨 3번을 눌렀을 때 ‘미’ 음을 Scale에 저장하는 프로그래밍 입니다.

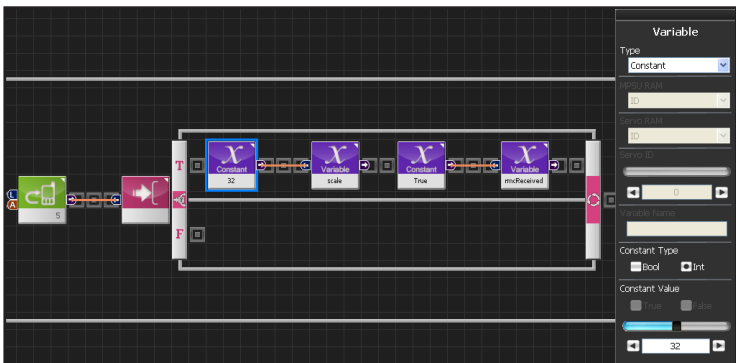
Scale = 29 번이 ‘미’ 음입니다.



22 4번 -> “파” 음

리모컨 4번을 눌렀을 때 ‘파’ 음을 Scale에 저장하는 프로그래밍 입니다.

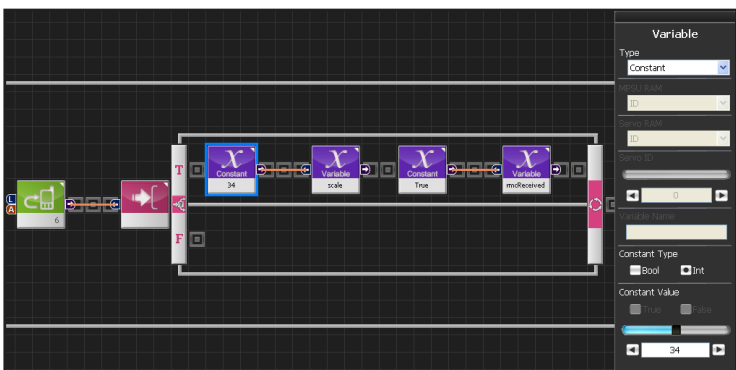
Scale = 30 번이 ‘파’ 음입니다.



23 5번 -> “솔” 음

리모컨 5번을 눌렀을 때 ‘솔’ 음을 Scale에 저장하는 프로그래밍 입니다.

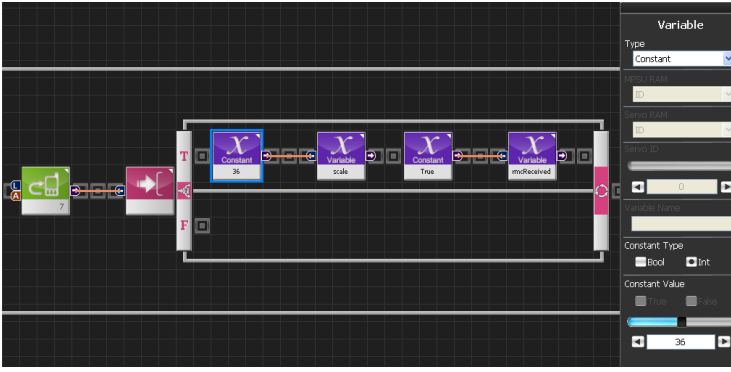
Scale = 32 번이 ‘솔’ 음입니다



24 6번 -> “라” 음

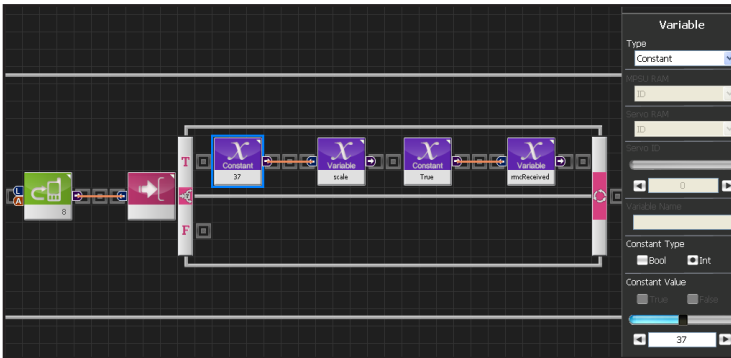
리모컨 6번을 눌렀을 때 ‘라’ 음을 Scale에 저장하는 프로그래밍 입니다.

Scale = 34 번이 ‘라’ 음입니다 .



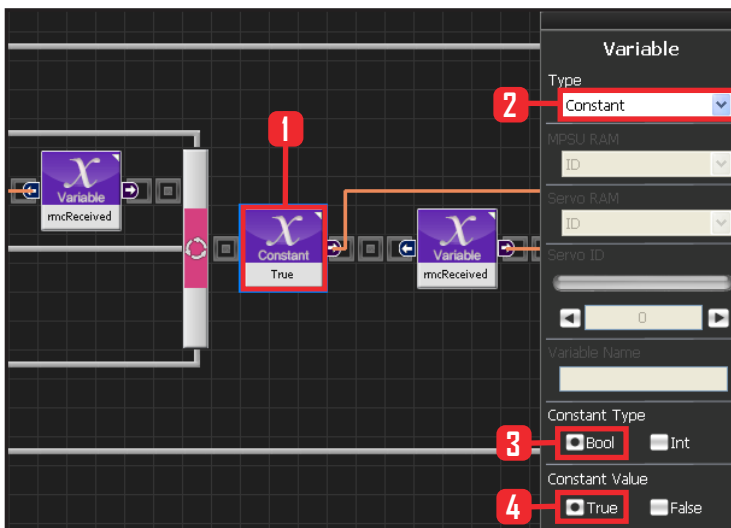
25 7번 -> "시" 음

리모컨 7번을 눌렀을 때 '시' 음을 Scale에 저장하는 프로그래밍입니다.
Scale = 36 번이 '시' 음입니다



26 8번 -> "도" 음

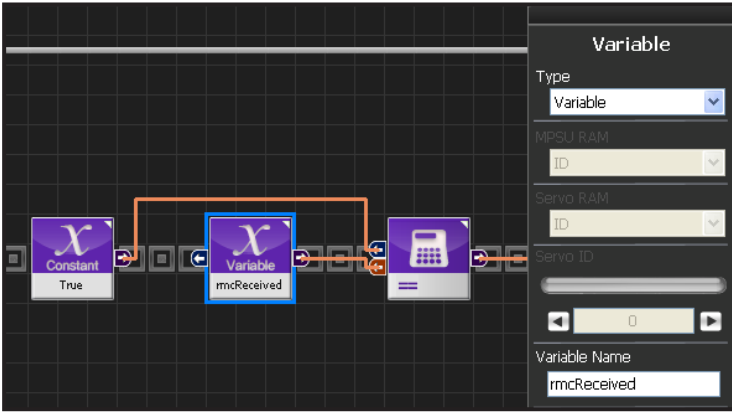
리모컨 8번을 눌렀을 때 '도' 음을 Scale에 저장하는 프로그래밍입니다.
Scale = 37 번이 '도' 음입니다.



27 rmcReceived 가 True 일 때

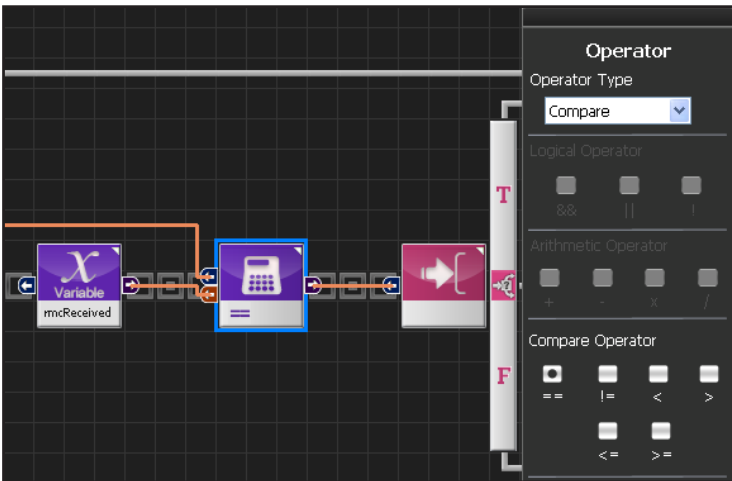
rmcReceived 가 True 일 때 앞서 음 높이 값을 저장한 scale 값을 note 에 넣어줘서 음이 나오게 합니다.

Data > Variable 모듈을 선택합니다.
Type : Contant 를 선택합니다.
Constant Type 은 Bool 로 설정합니다. Bool 은 참과 거짓을 나타내는 자료형입니다.
Constant Value : True 를 선택합니다



28 rmcReceived 가 True 일 때

rmcReceived 변수명은 동일합니다.



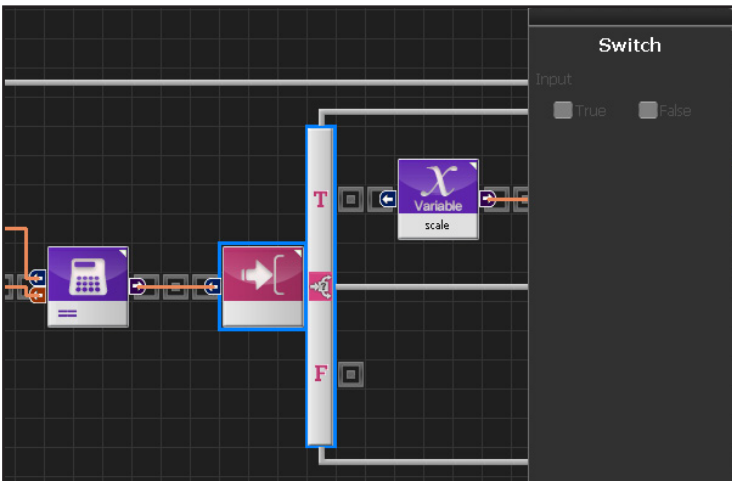
29 비교연산자 ==

Data > Operator 모듈을 선택합니다.

Operator Type : Compare 를 선택합니다.

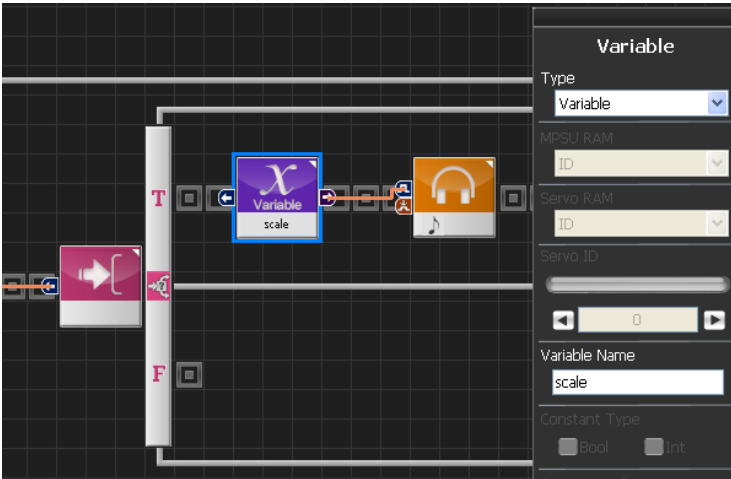
Compare Operator 는 == 를 선택합니다.

rmcReceived == true 라는 의미이며, "rmcReceived 가 true 와 같다" 를 나타냅니다.



30 Switch IF 분기문

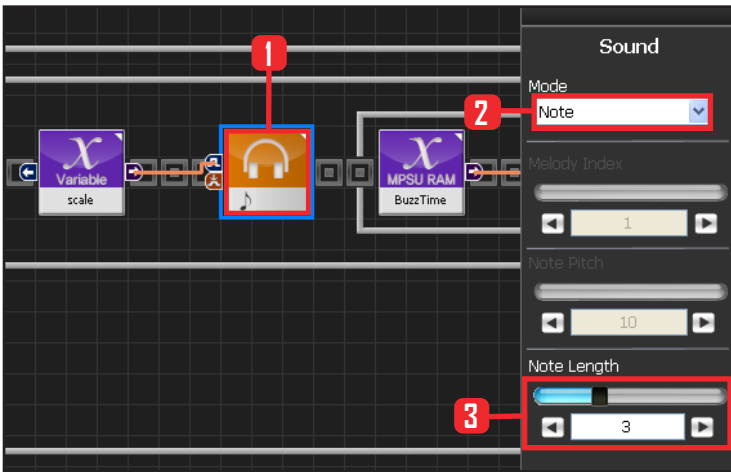
True 이면 실행합니다.



31 scale -> note

Scale 값을 Note 에 넣어주는 과정입니다.

변수 scale 모듈을 만듭니다.



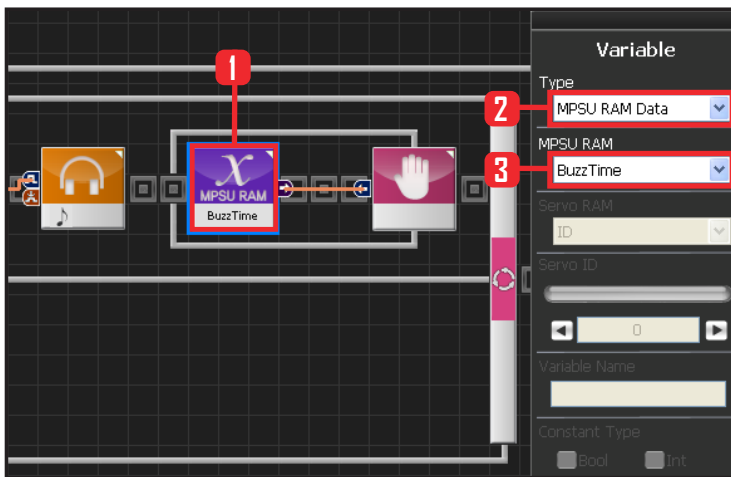
32 사운드 Play

Scale 값을 note 에 넣어줘 소리가 나오게 합니다.

Motion > Sound 모듈을 선택합니다.

Note Length : 3을 선택합니다. 8분음표를 뜻하며 153.6ms 동안을 합니다.

앞서 리모컨으로 부터 입력받은 버튼에 따라 scale 값을 다르게 저장하였고, 그 scale 값을 Note Pitch 넣어주면 거기에 해당하는 음계가 울립니다.



33 BuzzTime

MPSU RAM Data 에 BuzzTime 은 음이 울리고 있는지 여부를 판단하고 기다립니다.

BuzzTime은 버저가 울리기 시작할때 특정값이 된 후에 6.4ms마다 1씩 줄어듭니다.

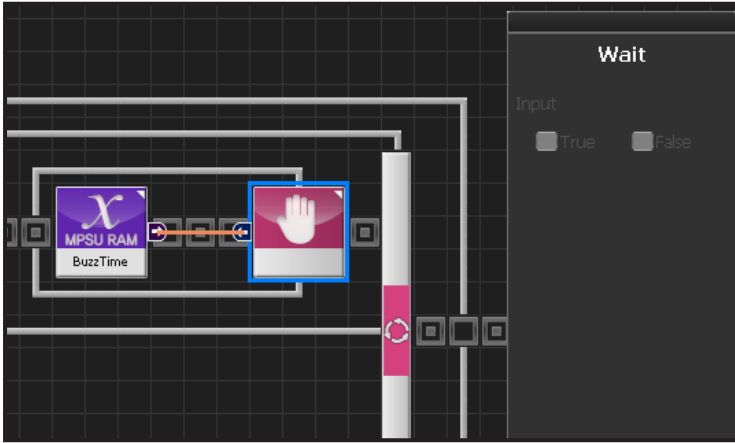
0이 아닐 때 버저가 울리고 있는 상태가 되고, 0일 때 버저가 안 울리고 있는 상태가 됩니다.

음표에 따른 초기 BuzTime 값은 두번째 페이지의 음길이 표에서 "Raw Data"를 참조하시면 됩니다.

Data > Variable 을 선택합니다.

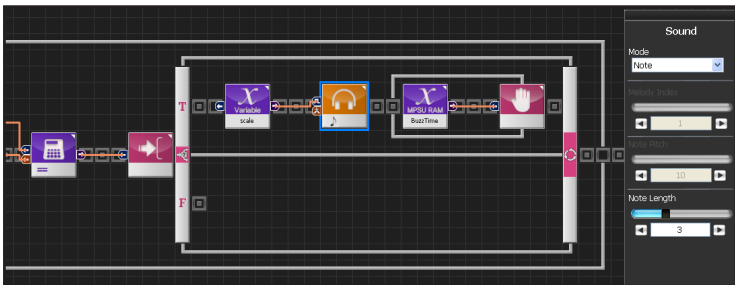
Type : MPSU RAM Data 를 선택합니다.

MPSU RAM : BuzzTime 을 선택합니다.



34 Wait

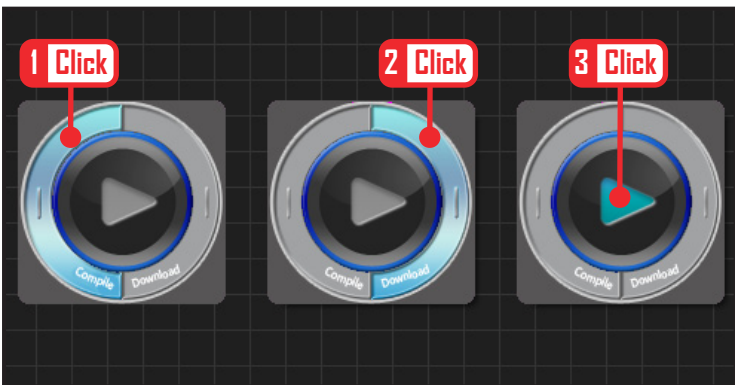
Buzztime이 0이 될때까지, 즉 소리가 끝날 때까지 기다립니다



35 음이 나오는 과정

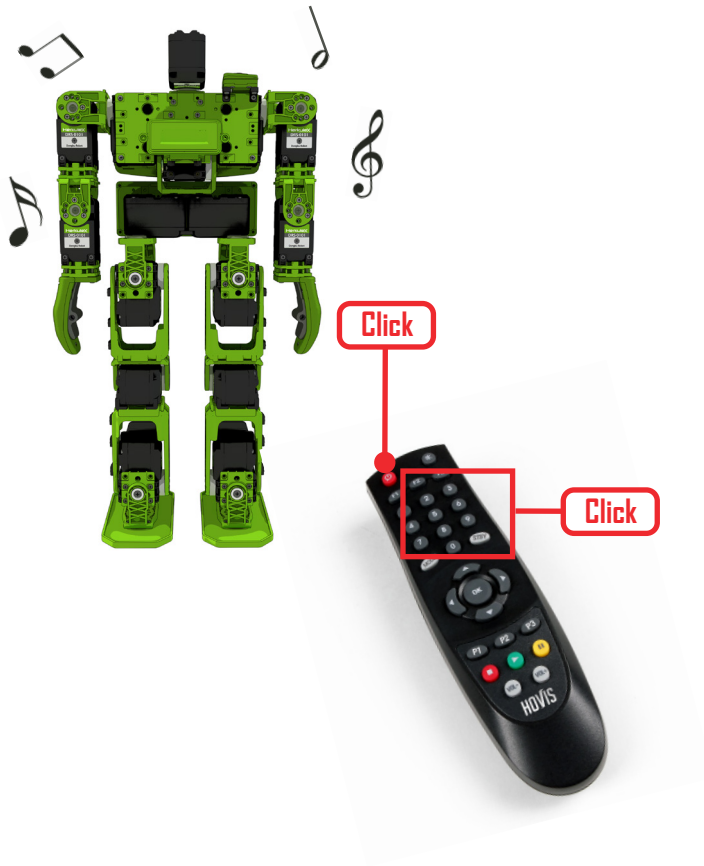
mcReceived 가 True일때 scale에 저장된 값을 Sound 모듈에 입력하여 Sound 모듈에서 해당음이 발생합니다.

그 음은 BuzzTime 으로 종료를 확인하고 다시 처음으로 돌아갑니다.



36 컴파일, 다운로드, 실행

왼쪽 클릭하여 컴파일 시킵니다. 에러가 없으면 오른쪽 클릭하여 로봇에 다운로드 시킵니다. 다운로드 완료되면 가운데 화살표 실행버튼을 눌러 로봇에서 실행시킵니다.



37 로봇동작

리모컨 1~8버튼을 누르면 음계를 실행하고, 전원 버튼을 길게 누르면 Task가 종료됩니다.