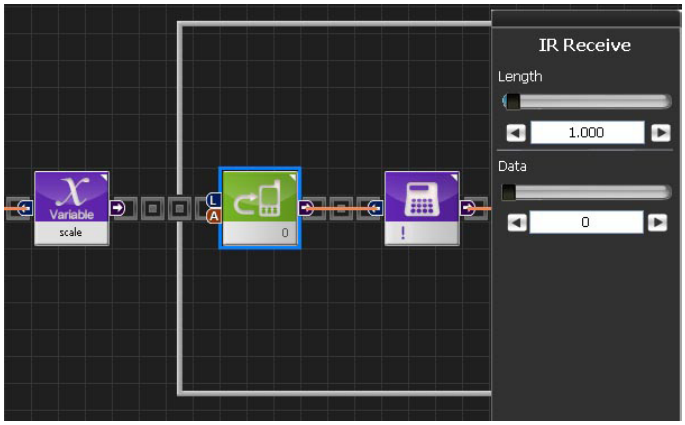
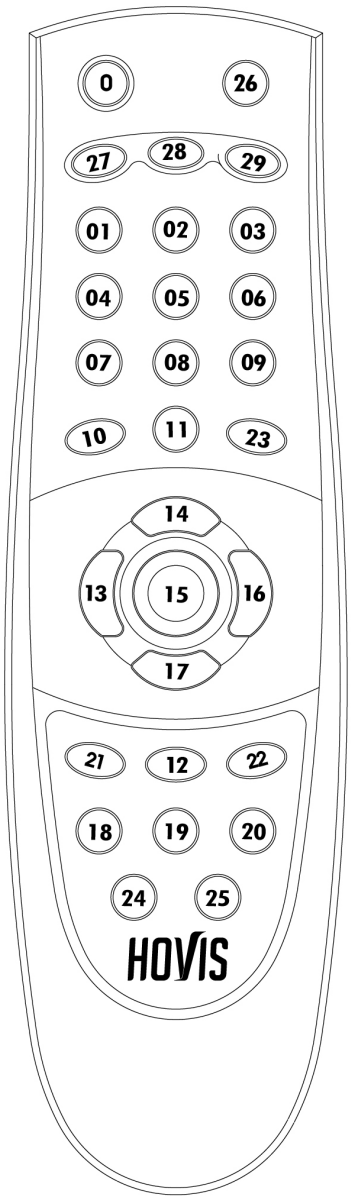
IRReceive, Sound & Motion Example Step by Step

(Explain by Sound examples, skip the explaination of motion examples, Data Match for Remote Controller)

Data figure from IR Recieve Module shall match the key from on the right side of remote controller.

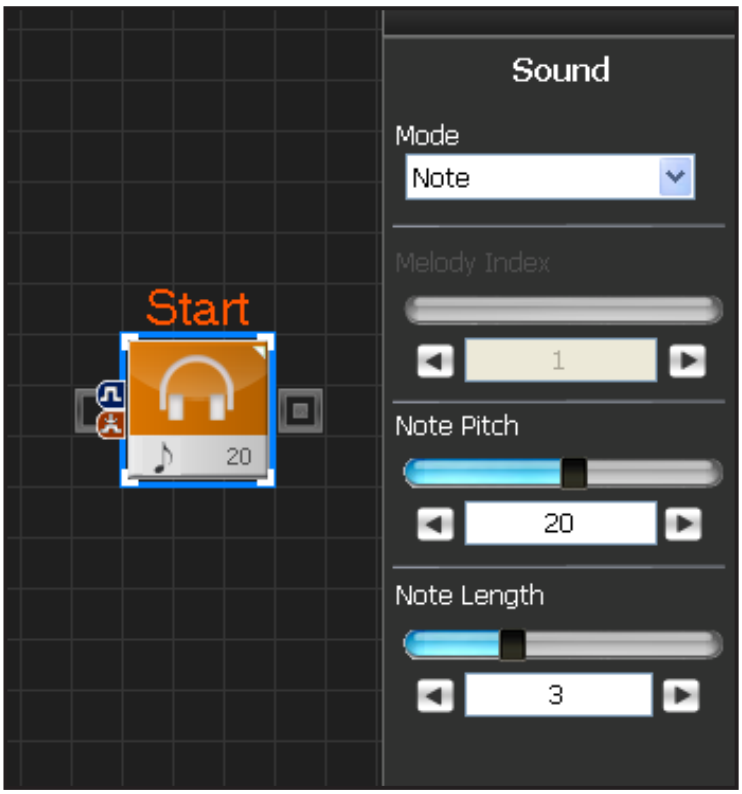




Example Description

This example associates remote control number button to a music note and outputs Do,Re,...Do (1~8) notes.

Note pictch is dependent on the value of the Note Pitch in Motion 〉Sound module. DRC controller has total of 38 pitches from 0~37 and it is able to ouptut total of 3 octaves.



00 Sound Property Window

Select Motion 〉Sound module.
Mode has Melody & Note. Melody selects and plays one of the saved edited notes.
Note Mode is selected to use the 36 note pitches. Refer to the table below

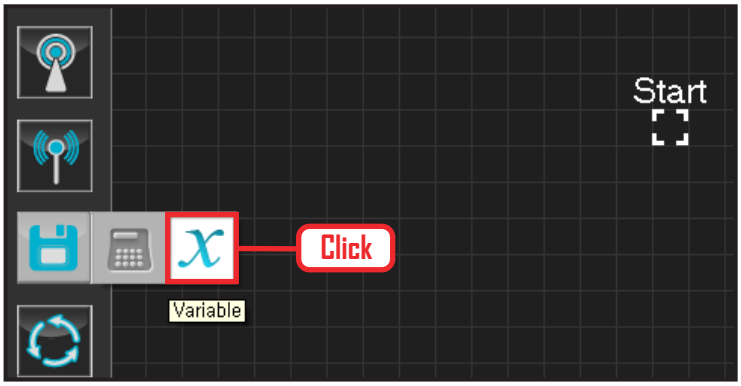
Note Pitch from 0~37 can be selected. Note pitches comprise total of of 3 octaves.
Note Length refers to the beat. Thirty-second note to the whole note can be selected.
Refer to the table below

Note Pitch

No.	0												
Note	NA												
No.	1	2	3	4	5	6	7	8	9	10	11	12	
Note	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si	
No.	13	14	15	16	17	18	19	20	21	22	23	24	
Note	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si	
No.	25	26	27	28	29	30	31	32	33	34	35	36	37
Note	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si	Do

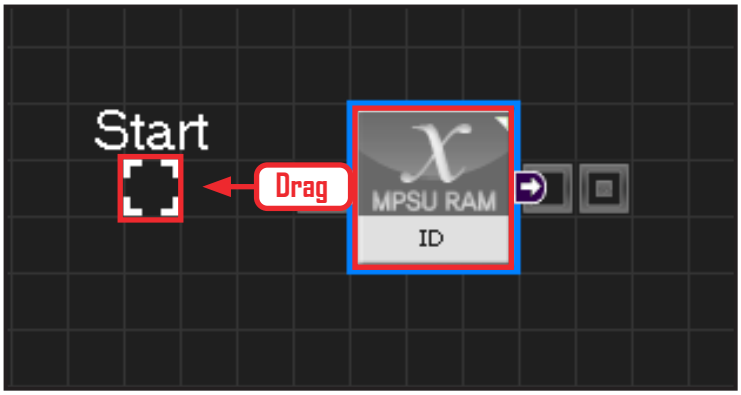
Note Length

No.	0	1	2	3	4	5	6	7	8	9
Raw Data	6	12	18	24	36	48	72	96	144	192
(ms)	38.4	76.8	115.2	153.6	230.4	307.2	460.8	614.4	921.6	1228.8
Note	32 note	16 note	16 dot note	8 note	8 dot note	4 note	4 dot note	2 note	2 dot note	Whole note



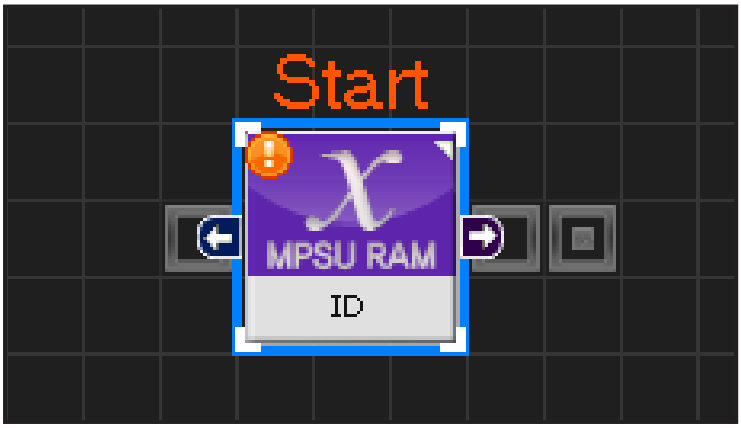
01 Assign Variable

Select Data 〉Variable module.



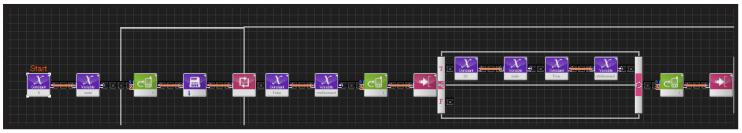
02 Start

Click and drag the connecting line located at left side of the module to the Start Point and dock



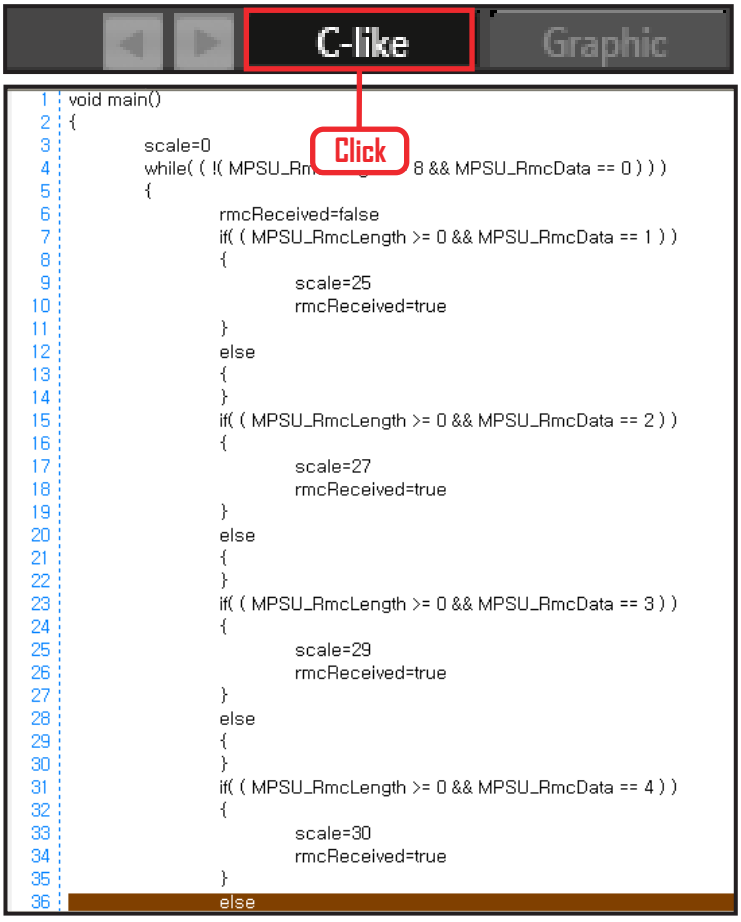
03 Start Programming

When the module and the Start Point is docked properly, module will become active and change color as seen in the photo to the left.This means programming has started..



04 Entire Program

Entire program using the remote control and the buzzer.

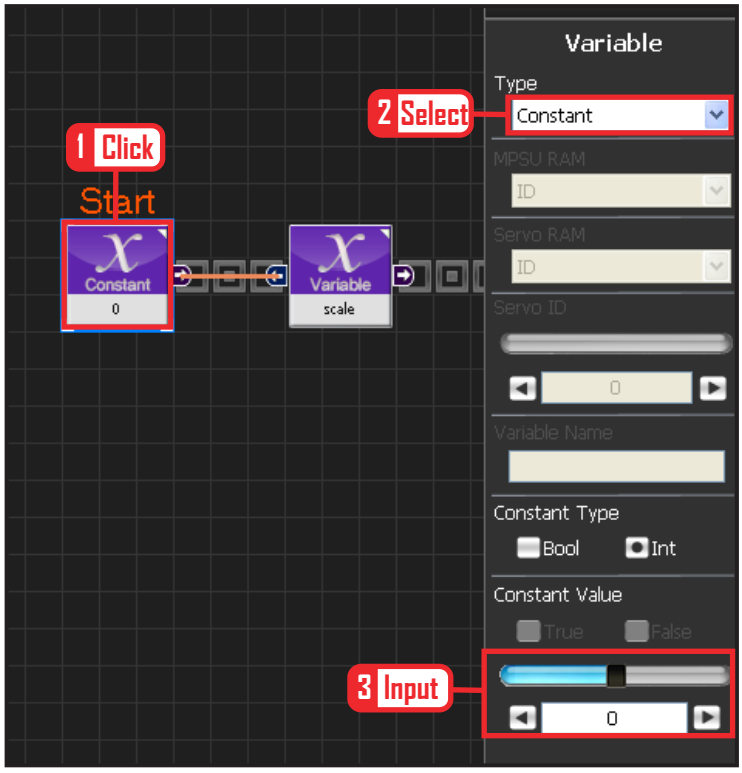


05 Viewing C-Like

Click the 'C–like' tab near the top right and task programming window will open as shown in the photo to the left. This is the task window of the entire program. Codes are very similar to the C language structure so studying the codes will help the user become familiar with the C language structure. Cursor will jump follwing the clicked module, making it easy to see the module changing to text.

```
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 Setup Constant

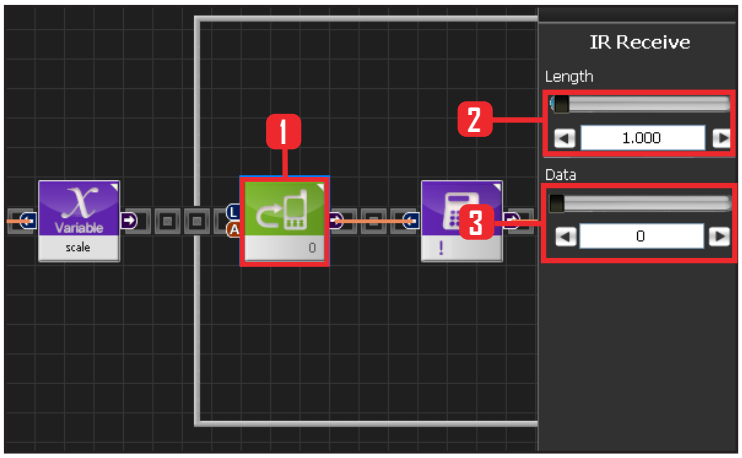
Declare variable of the scale to be played.

Select Data 〉Variable .
Select Type : Constant .
Set Constant Value : 0 .

07 Variable Name

Declare the name of the scale variable to be played.

Select Data 〉Variable .
Select Type : Variable .
Set Variable Name : scale .

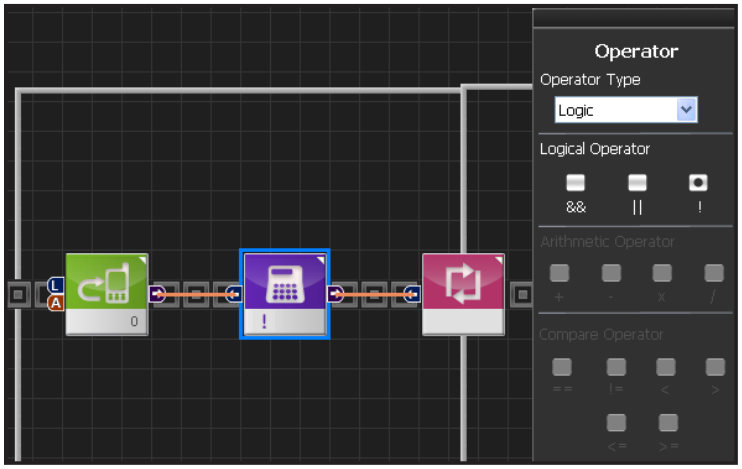


08 While Statement Exception

Exits if remote control button 0 is pressed loger than set time.

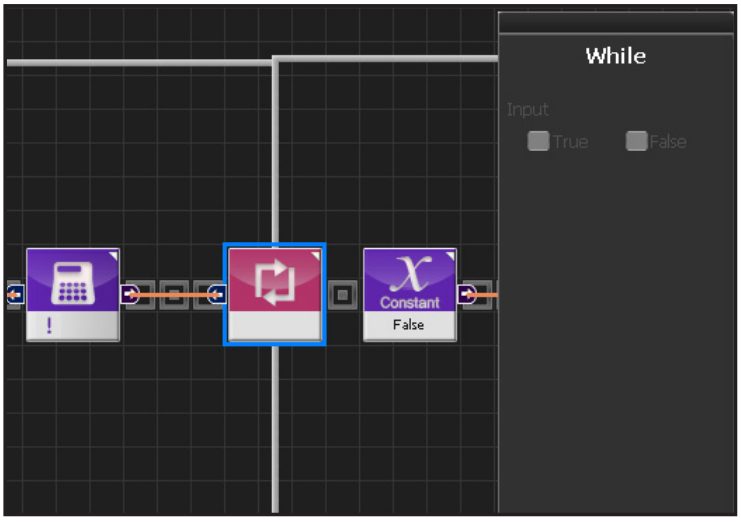
Select Communication 〉IRReceive module.
Set Length : 1.000 . 1s button press.
Set Data : 0 . Power button press.

When the power button is pressed longer than 1s, output of the module is True. False if less than 1s.



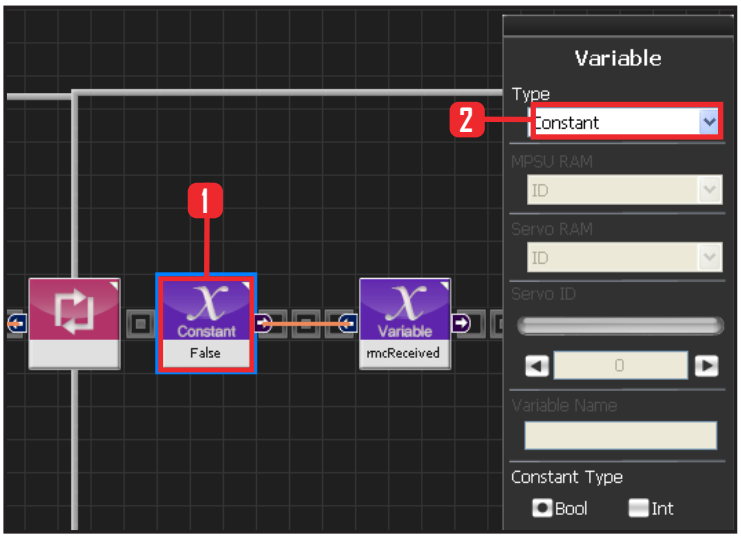
09 Setup ! operator

! converts true / false value to opposite. Output value of IRReceive module is converted to opposite value and used as input value of the while statement.



10 While Loop

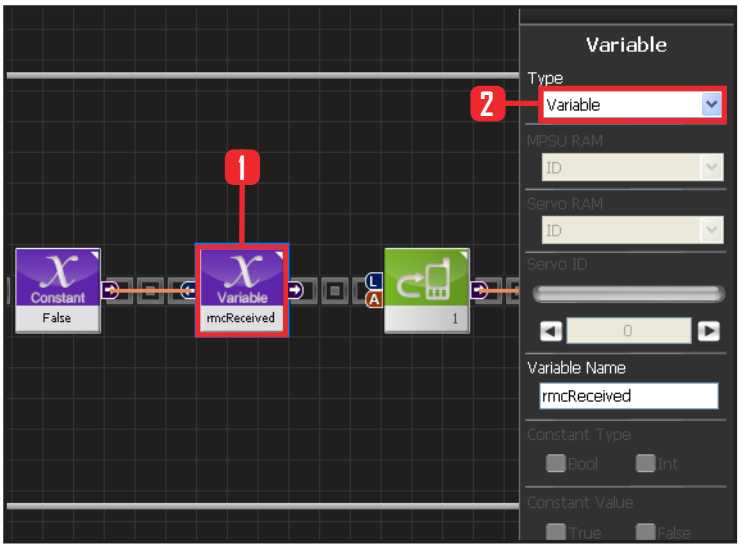
Repeat depending on previous condition.
If True, continue to repeat next step.By going through the ! operator, repeat if the ouput value of the IRReceive module is false, exit loop if true.
Exit loop if the power button is presed longer than 1s.



11 Initialize Remote Control Input Variable.

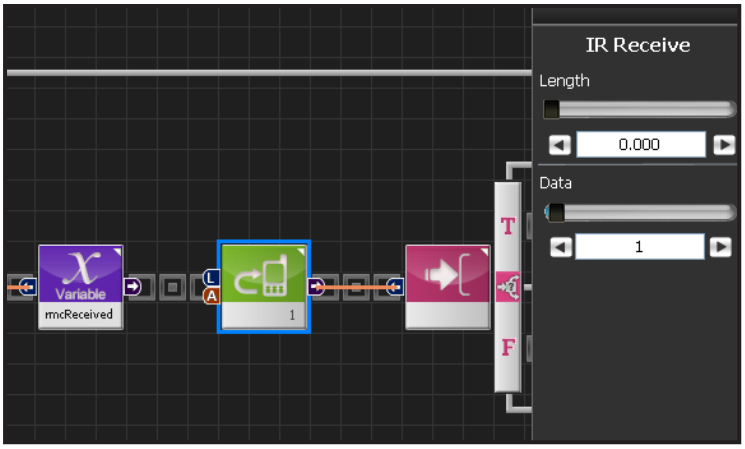
Select variable showing that remote control input was received.

Select Data 〉Variable module.
Select Type : Contant .
Select Constant Type: Bool: True or False data type
Select Constant Value : False



12 Remote Control Input Initial Variable.

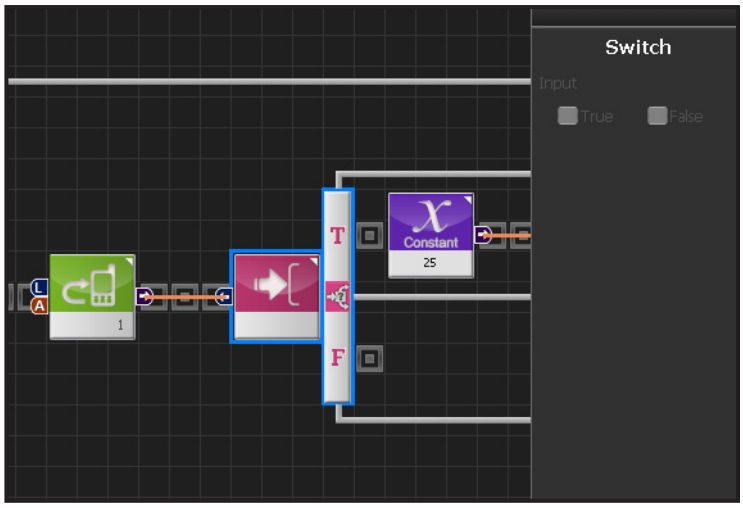
Select Data 〉Variable .
Select Type : Variable .
Variable Name : rmcReceived
rmcReceived is a variable showing that remote control button 1~8 input was received within the loop. Intitalized as False at beginning of the loop.
Play note if the checked value towards the end of the loop is True.



13 Remote Control Button 1

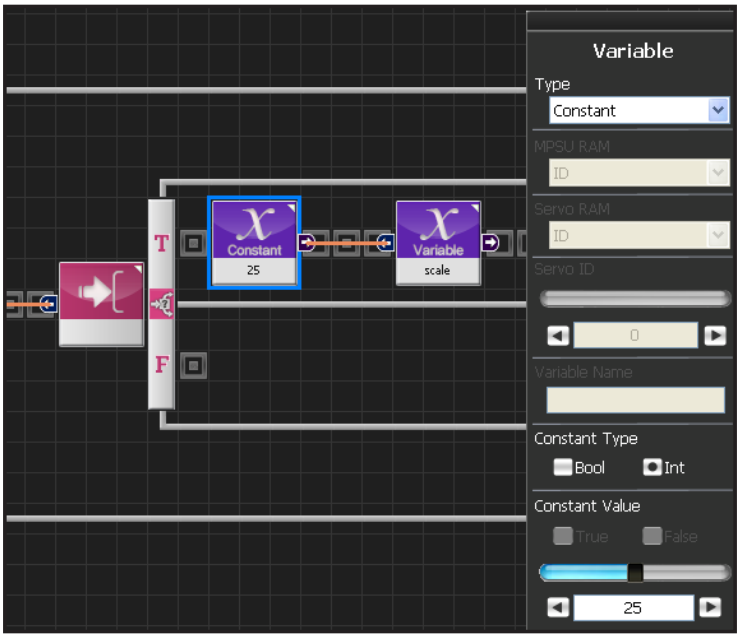
Check if remote control button 1 was pressed.

Select Communication〉IRReceive module
Set Length : 0.000 .
Set Data : 1 . Refers to Button1.



14 IF Conditional Statement

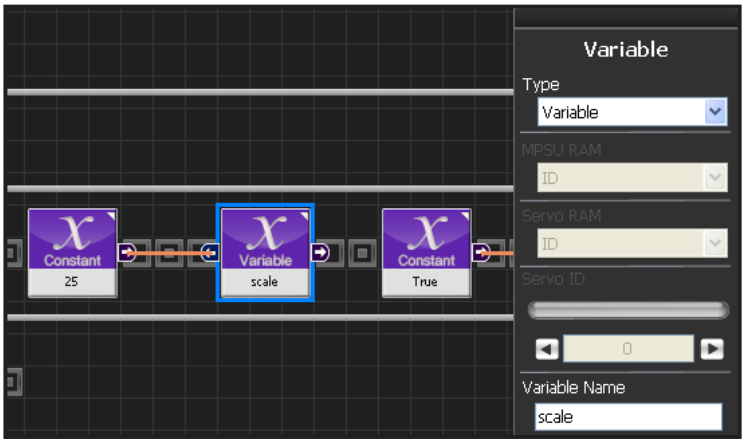
Run if True.



15 Save "Do" Note

As explanined previously, Note Pitch (3 octaves) number 25 referst to 'Do' note.
Change the Scale value to 'Do'

Select Data〉Variable module.
Select Type : Contant
Select Constant Type: int.
Set Constant Value : 25 . 25 refers to "Do".

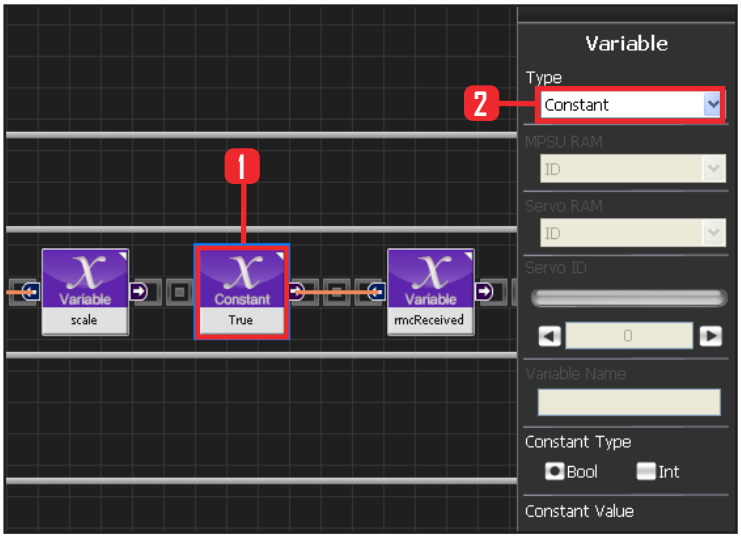


16 scale

Declare variable name of the scale to be playes as Scale.

Select Data 〉Variable .
Select Type : Variable .
Set Variable Name : scale .

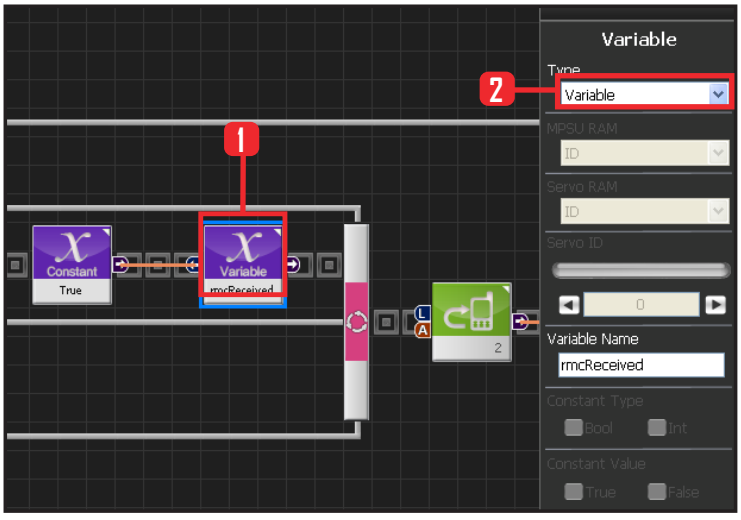
Receive previous constant value 25 using input connecter .



17 Save Remote Control Input Confirm Valule

If rmcRecieved value is True, it denotes one of the remote control button (1~8) was pressed.

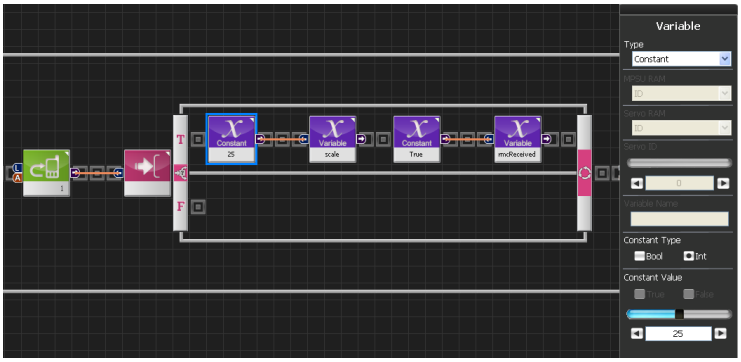
Select Data 〉Variable module.
Select Type : Contant .
Select Constant Type : Bool .
Select Constant Value : True .



18 Save Remote Control Input Confirm Value

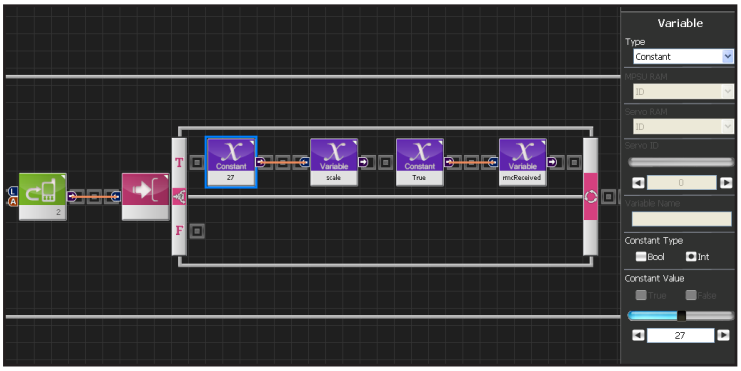
Select Data 〉Variable .
Select Type : Variable .
Set Variable Name : rmcReceived.

Receive previous connstant value True using input connertor.



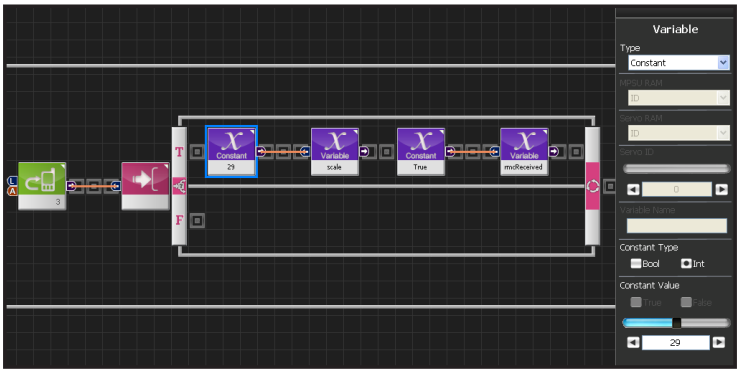
19 1 -> "Do" Note

Program saves note 'Do' in the scale when reomote control button 1 is pressed.



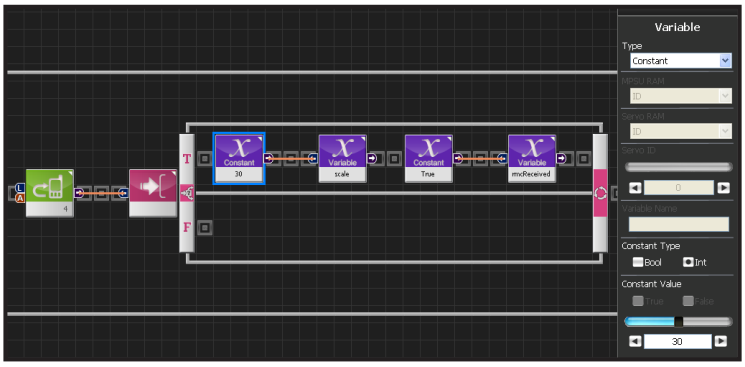
20 2 -> "Re" Note

Program saves note 'Re' in the scale when reomote control button 2 is pressed.
Scale = No 27 is 'Re' note.



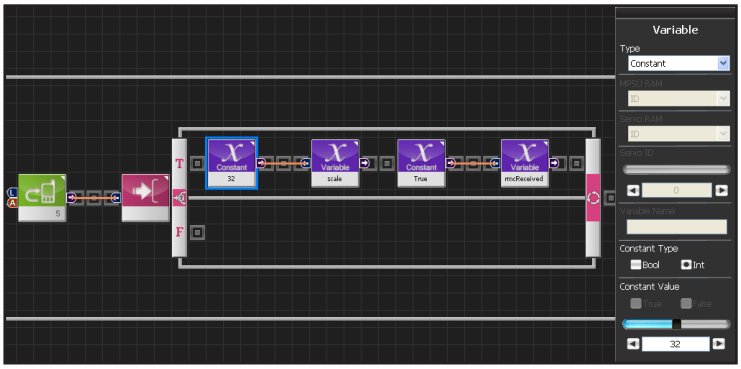
21 3 -> "Mi" Note

Program saves note 'Mi' in the scale when reomote control button 3 is pressed.
Scale = No 29 is 'Mi' note.



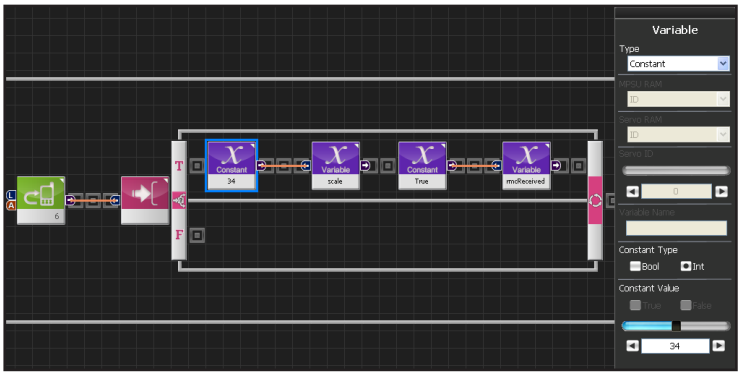
22 4 -> "Fa" note

Program saves note 'Fa' in the scale when reomote control button 4 is pressed.
Scale = No 30 is 'Fa' note.



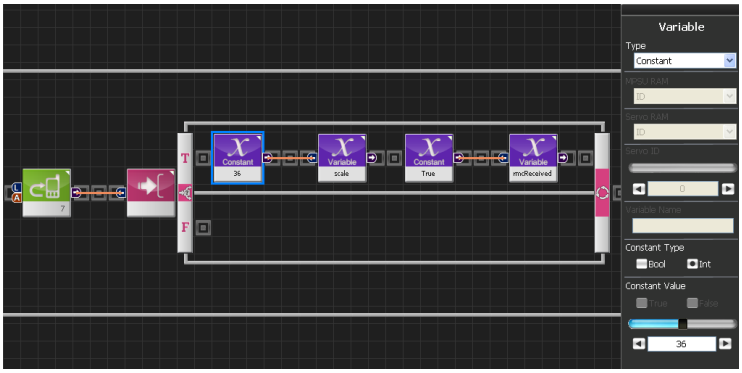
23 5 -> "Sol" Note

Program saves note 'Sol' in the scale when reomote control button 5 is pressed.
Scale = No 32 is 'Sol' note



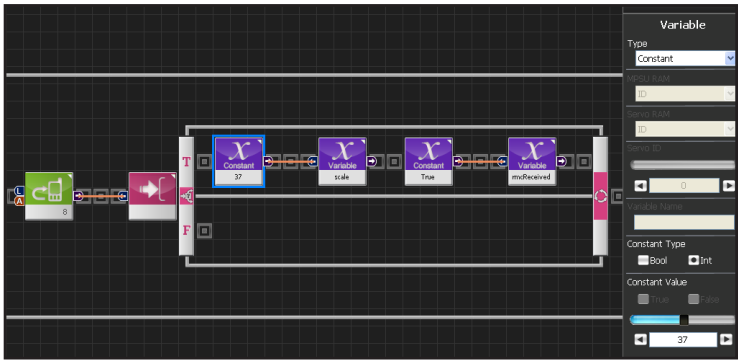
24 6 -> "Ra" Note

Program saves note 'Ra' in the scale when reomote control button 6 is pressed.
Scale = No 34 is 'Ra' note .



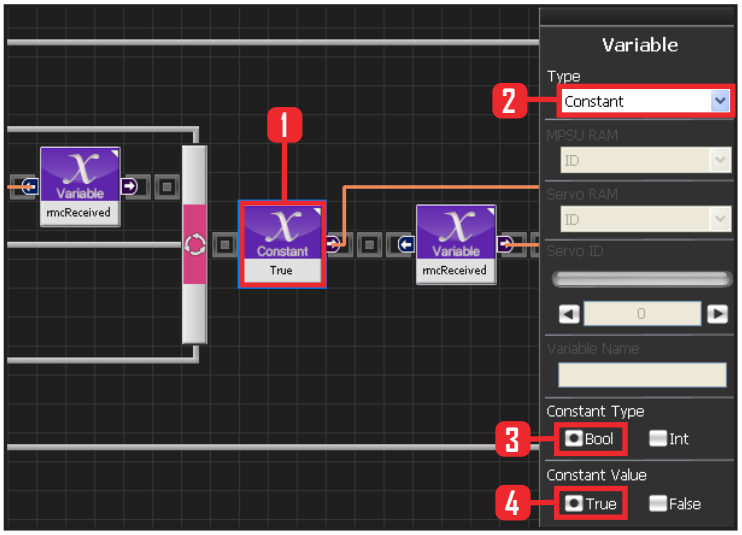
25 7 -> "Si" Note

Program saves note 'Si' in the scale when reomote control button 7 is pressed.
Scale = No 36 is 'Si' note.



26 8-> "Do" Note

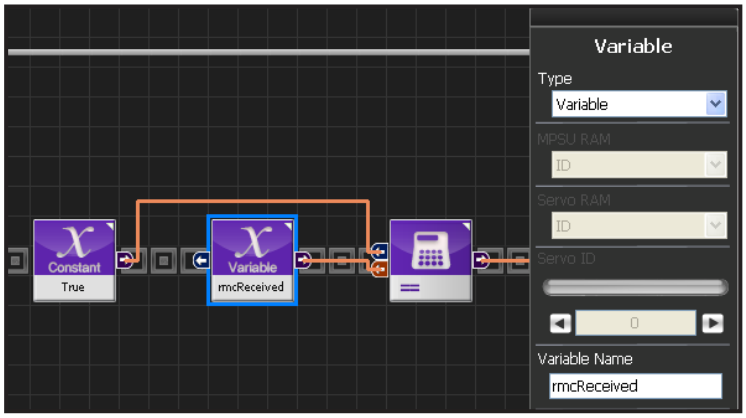
Program saves note 'Do' in the scale when reomote control button 8 is pressed.
Scale = No 37 is 'Do' note .



27 Whe rmcReceived is True

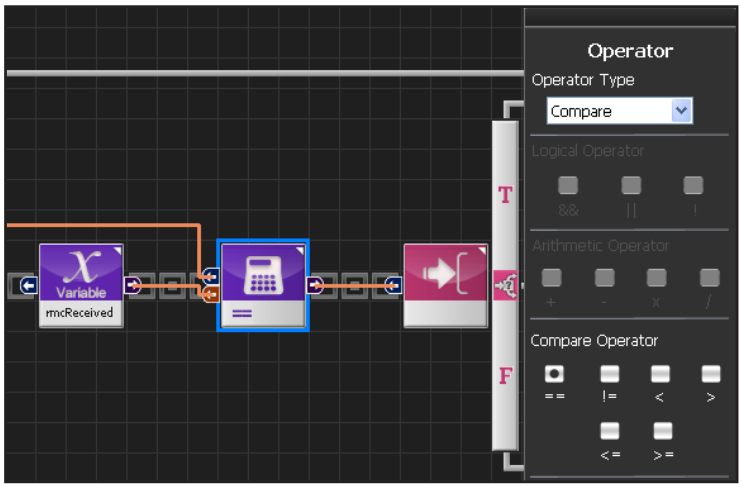
When rmcReceived is True, input saved scale value where pitch value was previously saved into note to ouput note.

Select Data 〉Variable module.
Select Type : Contant .
Select Constant Type: Bool.
Select Constant Value : True.



28 When rmcReceived is True

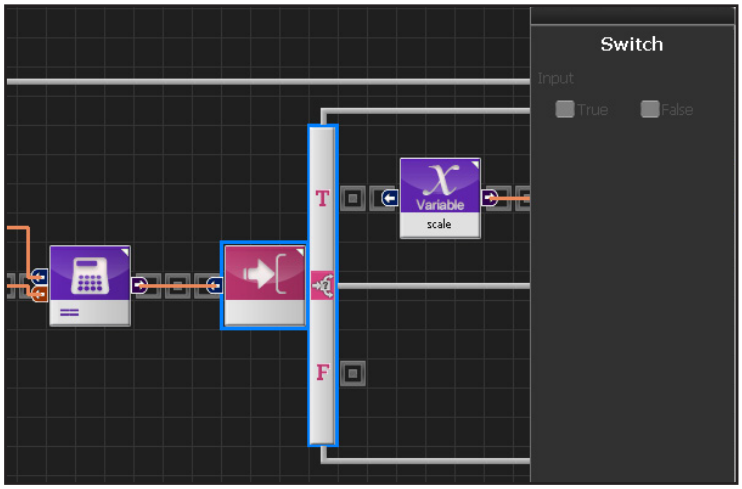
rmcReceived variable name is identical.



29 Comparison Operator ==

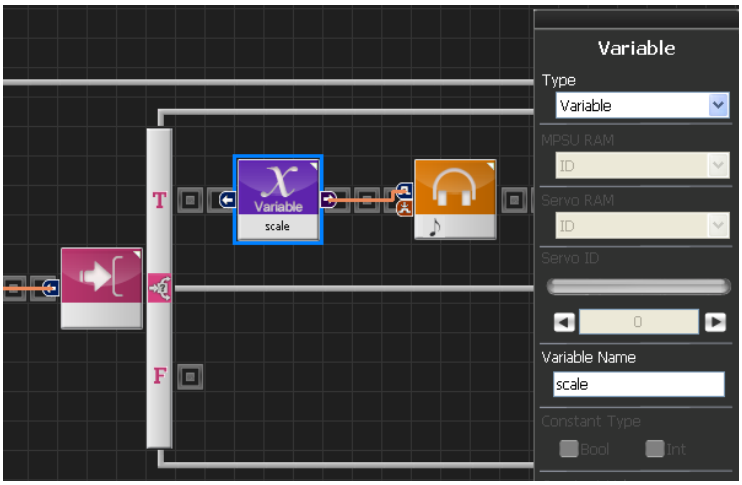
Select Data 〉Operator module
Select Operator Type : Compare .
Select Compare Operator: == .

rmcReceived = = refers to true ,
shows “rmcReceived is equal to true .



30 Switch IF Conditional Statement

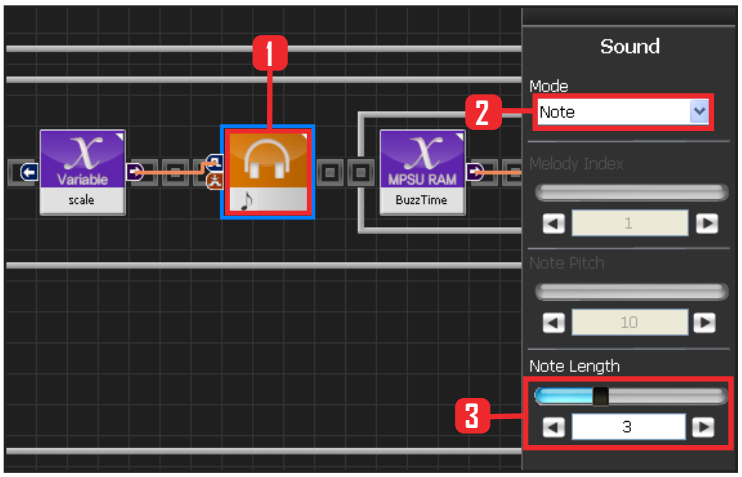
Run if True.



31 scale -> note

Input Scale value into Note.

Make variable scale module.

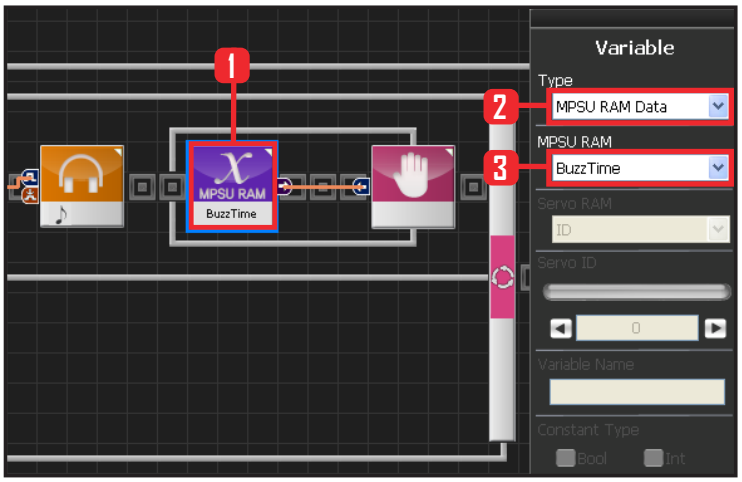


32 Sound Play

Input Scale value into note to play sound.

Select Motion 〉Sound module.
Set Note Length : 3. detnotes eighth note. Lasts 153.6ms .

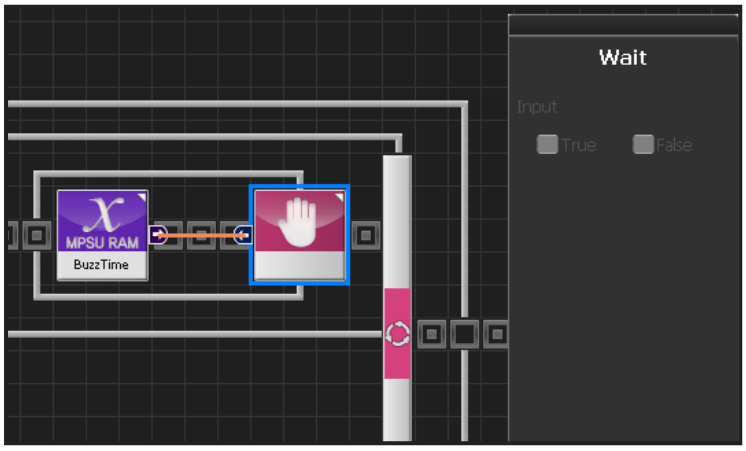
Different scale values were saved depending on the input from the remote control buttons. When the scale value is recevied by Note Pitch correspoding note will play.



33 BuzzTime

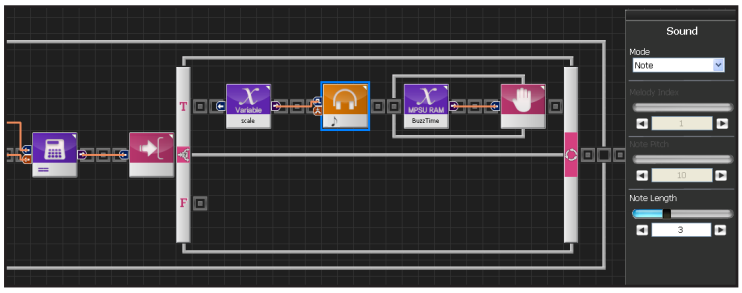
Buzz Time in MPSU RAM Data decides if the note is playing and waits.
When buzzer starts to sound, BuzzTime acquires certain value which decreases by 1 every 6.4ms. If the value is other than 0, buzzer is still sounding and if the value is 0, buzzer has stopped. Refer to 'Raw Data' in note length table for initial BuzzTime values.

Select Data 〉Variable .
Select Type : MPSU RAM Data .
Select MPSU RAM : BuzzTime .



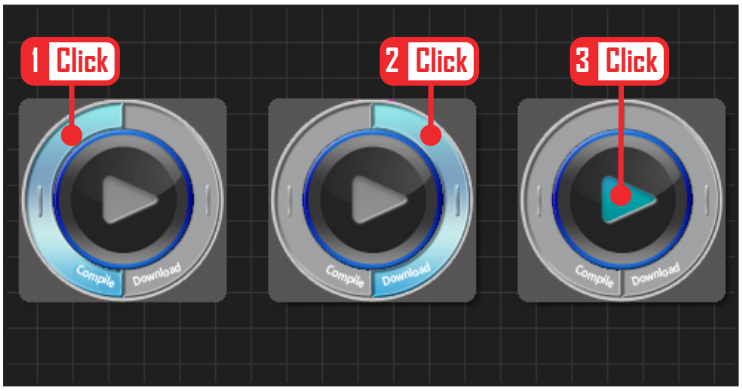
34 Wait

Wait untill Buzztime value becomes 0, In othe words, wait untill the sound ends.



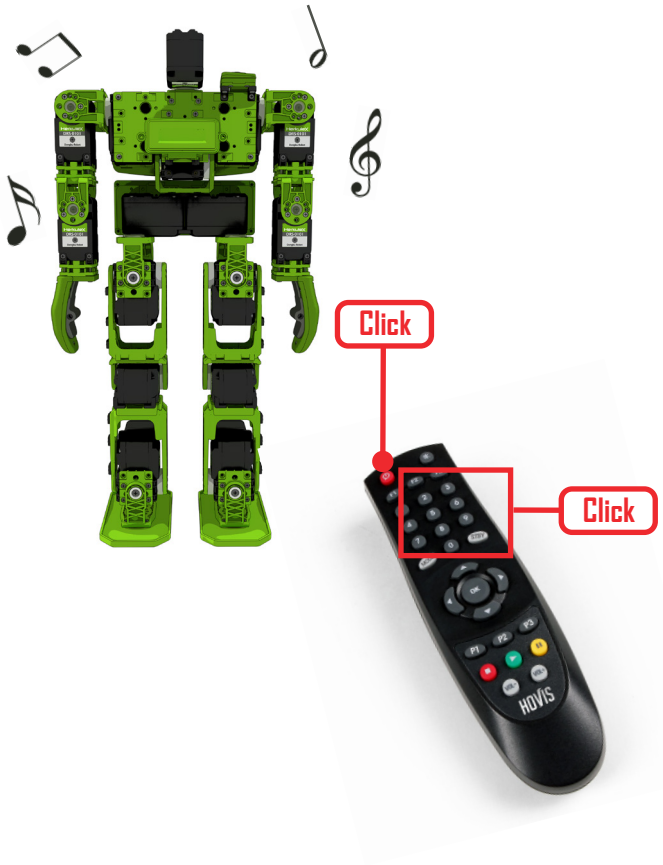
35 Note Output Process

When rmcReceived is True, value saved in scale is used as input to Sound module which then outputs corresponding note.
BuzzTime to checks the end of the note and goes back to the begining.



36 Compile, Download, Run

Click 'Compile'. Click 'download' on the right if there is no compilation error. Download to robot. Click 'Run' button (Arrow button) after the download..



37 Robot Motion

Press Remote control buttons(1~8) to play notes. End task by pressing the power button for more than 1s.