

## Software Nodes Card



**Nodes Name:** Compare

**Explanation:** Compare the inputting number with a pre-configured number.

**How it is used:** when you want something to happen if some value is above, below or equal a certain number. For example:

- 1) If the environment is overheated (temperature above a certain number), sound the alarm;
- 2) If the counter counts to 3, show a smiley face.

**How it works:**



The input is the value to be compared, Pick an operator “<”, “>”, “=”, “≠”, “≤”, “≥”.  
If the value satisfies the condition, the node will output “yes”; otherwise, it will output “no”.



When the temperature is over 30 (degrees Celsius), sound the buzzer.

**Tips:**

- 1) If the input is a string, the node will try to convert it to a number; if failed, the output will be “no”.
- 2) If the input is an object, the output will always be “no”.
- 3) A hanging input yields “no”.

## Software Nodes Card



**Nodes Name:** Number

**Explanation:** Provide a number as a constant, or under a certain condition.

**How it is used:** when you want something to happen if some value is above, below or equal a certain number. For example:

- 1) Set the parameter of another node. E.g., the rotation speed of a motor;
- 2) Let another node's parameter change to a certain value when something happens. E.g., when the heat is up, the fan motor rotates at a certain (higher) speed.

**How it works:**



The number set in the config panel will be sent to the output. If an input is provided, the value is sent only if the input is “yes”; otherwise the output will be “no”.



Example: set the speed of the DC Motor Drive to 50



Example: When the value of the light sensor is above 50, set the motor speed to 70.

## Software Nodes Card



**Nodes Name:** Compute

**Explanation:** Make a certain arithmetic (+, -, \*, /) operation on the input.

**How it is used:** If you want your creation reacts to the slightest change in the room temperature, or you want the car moves, but have more juice when the color is blue, the “Compute” node comes to help. It can: - Amplify the input to a certain factor - Give an initial value of input by adding it to a constant number - Count numbers in a cycle by using the “mod” operator.

**How it works:**



Pick an operator (“+”, “-“, “\*”, “/”) and an operand (the number you want to add or subtract with). The output will be the input compute with the configured operand.



When the temperature is over 30 (degrees Celsius), sound the buzzer.

**Tips:**

- 1) If the input is a string, the node will try to convert it to a number; if failed, the output will be “no”.
- 2) If the input is an object, the output will always be “no”.
- 3) A hanging input yields “no”.

## Software Nodes Card



**Nodes Name:** Interval

**Explanation:** Flips the output repeatedly, at a certain interval.

**How it is used:** The output of the interval node flips between “yes” and “no” by itself. It is useful when:

- 1) You want to make a blinking light or a jiggling robot;
- 2) You want to do something periodically, like taking a photo every 30 seconds.

**How it works:**



In the config panel, choose the how many seconds before the result flips.



Example: making a blinking light



Example: making a “beep-beep” alarm when overheat.

## Software Nodes Card



**Nodes Name:** Not

**Explanation:** YES to NO and NO to YES.

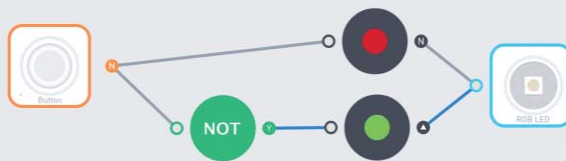
**How it is used:** Logical “NOT” stands for “if not”, “else” or “otherwise”. It flips the input from “yes” to “no” and “no” to “yes”. Useful in cases like:

- 1) Do something when the button is “NOT” pressed;
- 2) If the Interval node is flipped to no, do something else.

**How it works:**



Just attach the input, and get the opposite result.



Example: if the button is connected, turn red; otherwise, turn green.



Color cycles between red and green.

## Software Nodes Card



**Nodes Name:** And

**Explanation:** YES if all inputs are YES.

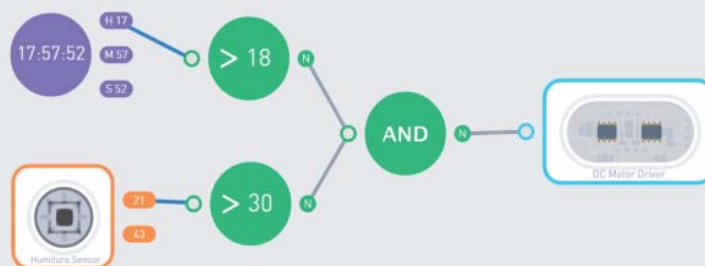
**How it is used:** The logical “AND” node outputs “yes” only when all the input sources say “yes”. It can be used in situations like:

1) If the room temperature is high “AND” it is past 6 o’clock, turn on the fan;

**How it works:**



The “AND” node says “yes” when all the sources of input are considered “yes”.



On a hot evening (temperature is over 30 and current hour is over 17, 5pm in the evening), turn on the fan (hooked up to a motor).  
Color cycles between red and green.

## Software Nodes Card



**Nodes Name:** Or

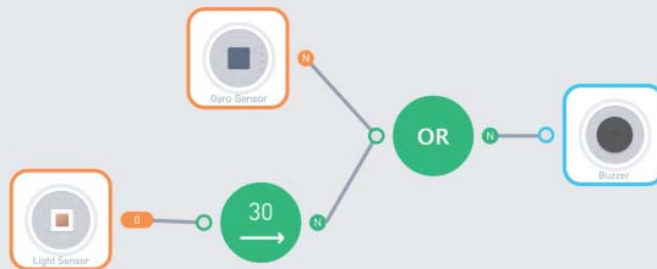
**Explanation:** YES if any of the inputs is YES.

**How it is used:** The logical “OR” node outputs “yes” if any of the input sources say “yes”. It can be used when: - If there is light (the drawer is open), sound the alarm; If the gyro is shaken (the creation is moved by somebody), also sound the alarm.

**How it works:**



The “OR” node says “yes” whenever a source of the input is considered “yes”.



Example: an alarming device that alarms when exposed to light (light sensor value > 30), or if it is moved (the gyro is shaken).

**Tips:**

1) In fact, you do not need an “OR” node very often because an “OR” logic is placed inside the input of most nodes.

## Software Nodes Card



**Nodes Name:** Toggle

**Explanation:** Flip between YES and NO when activated.

**How it is used:** The Toggle node flips between “yes” and “no”. You can use it to: - Convert a button to a switch. Push it, lights on (without the needs of keeping your finger on the button); push again, lights off.

**How it works:**



When the input changes from “no” to “yes”(we call it a “rising edge” in electric engineering), the output will flip between “yes” and “no”.



Example: use a button as a switch



## Software Nodes Card



**Nodes Name:** Counter

**Explanation:** Plus one when activated.

**How it is used:** The “Counter” node keeps a number for counting; it goes up each time the input flips from “no” to “yes”. This is useful when you want to:

- 1) Keep a number of something, like how many times the gyro is shaken;
- 2) Make a stopwatch: count how many seconds did some event last;
- 3) Make a timer: let something happen after a certain time period is passed.

**How it works:**



When the input changes from “no” to “yes”(we call it a “rising edge” in electric engineering), the number displayed in the node will plus one, and goes to the output. If you push the “reset” button, the number will be reset to zero; the same will happen if the “RESET” input changes from “no” to “yes”.



Example: count the length of time when the room filled with people (using the PIR “people sensor”). The result is expressed in seconds.



Example: making a countdown, use the COMPUTE+ node.

## Software Nodes Card



**Nodes Name:** Hold

**Explanation:** Keep the input unchanged for a period of time.

**How it is used:** You can let buzzer sing a song whenever you shake the gyro. But as soon as the gyro is not shaken, the song will instantly stop. To let the song keep playing, you need to “HOLD” the “gyro is shaking” input for a period of time. This is when HOLD comes into the scene. It can:

- 1) Keep the input value, until another not “no” value comes in;
- 2) Keep the input value unchanged for a period of time;
- 3) Let the input change gradually.

**How it works:**



You can choose one of the three modes from the config pane.

Hold

Select

hold until change hold for time change slowly

Time

3

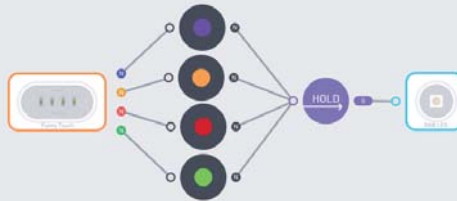
Confirm

NTROLS TIME ADVANCED

## Software Nodes Card



The first, Hold until change: the input will be kept until another input comes in (such as a sad face replaces a smiley face). In other words, the output could be any value other than “no”.



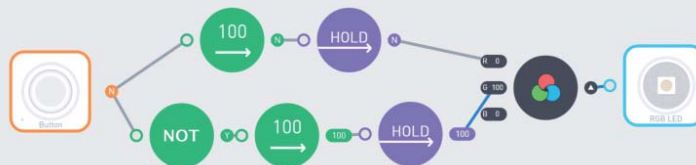
Example: count the length of time when the room filled with people (using the PIR “people sensor”). The result is expressed in seconds.

The second, Hold for time: the input will stay the same for a period of time. In this period, any other input values will be ignored. If the hold period has been passed and the input is still “no”, the output will be set to “no”.



Example: An alarming device that beeps 3 seconds after light detected.

The third kind, Change Slowly: the output will change according to the input, but if the input is a number, every second the change will be no more than the specified number.



Example: When the button is pressed, the light turns from green to red slowly (The number of the HOLD node’s config is set to 3).

## Software Nodes Card



**Nodes Name:** Delay

**Explanation:** Make it happen after some time.

**How it is used:** Delay node will take an input, and send it to the output after a set period of time. It is useful when:

- 1) You want something happens after an event but after a period of time;
- 2) You want something happens after an event, and after a period of time, some other thing happens after an event.

**How it works:**



Every input will be sent to the output after a certain period of time (provided in the config panel).



Example: When the button is pressed, rotate the motor, then rotate it reversely after 1 second.

**Tips:**

- 1) An easy way to understand the “DELAY” node is: the output is always certain seconds behind the input.

## Software Nodes Card



**Nodes Name:** Average

**Explanation:** Average over a period of time.

**How it is used:** Imagine you want to build a device that alarms the mother whenever the baby cry, using the movement value (acceleration) detected by the gyro sensor. But if the alarm sounds every time there is little disturbance of the sensor (e.g. when the baby flips his/her arm), the mother will be crazy. Using the “average” node will let the result reflects the average of the input value over a period of time, filtering out noise data.

**How it works:**



The result will be the average value of the input over a period of time. You can set the period of time (engineers call that sampling window) you want to perform the average in the config panel.



Example: a “baby monitor” that alarms the mother when the baby cries (which makes the gyro shaken for a period of time. The Y Acceleration is used to represent shaking).

**Tips:**

1) A smaller sampling window means the output is more sensitive to the change of the input.

## Software Nodes Card



**Nodes Name:** Today

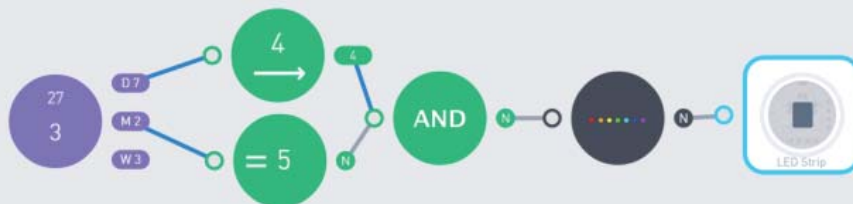
**Explanation:** Get the date (year, month, day) today.

**How it is used:** Node “TODAY” outputs today’s month, the day of the month, and day of the week. It can be used to: - Do something on a specific day.

**How it works:**



There are 3 outputs in this node, and the meaning is straightforward: the day of the month(D), the month(M), and the day of the week(W) of today.



Example: light up the light strip on May the 4th.

## Software Nodes Card



**Nodes Name:** Now

**Explanation:** Get the hour, minute, and second now.

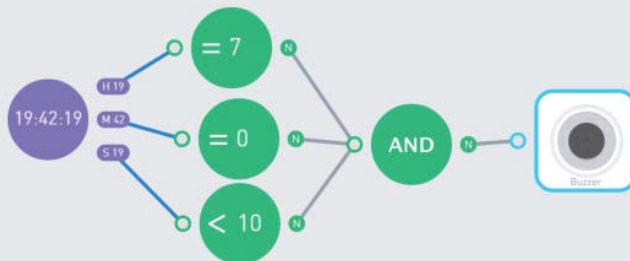
**How it is used:** NNode “NOW” outputs the current time, in the format of hour, minutes, and seconds. It can be used to:

- 1) Do something at a specific time of the day;
- 2) Do something every second, every a few seconds or every minute.

**How it works:**



There are 3 outputs in this node, and the meaning is straightforward: the hour(H), the minute(M), and the second(S) of the current time.



Example: Example: an alarm clock that buzzes for 10 seconds at 7:00.

## Software Nodes Card



**Nodes Name:** Pulse

**Explanation:** Make the output cycles over time.

**How it is used:** The PULSE node's output changes with the time according to the waveform specified in the config panel. It can be used to:

1) Make a "breathing" light/sound/movement effect; make the movement looks smooth;

**How it works:**



In the settings panel, you can set the following parameters of the pulse generated:

- 1) **Waveform:** it can be "sin", "square" and "triangle". "Sin" waveform looks like breathing: it slows down at the top and bottom value. "Square" acts similar to the "INTERVAL" note, it turns on and off at each time period. "Triangle" is steadier (linear) when changing, but drives in the opposite way sharply when it reaches its maximum and minimum value.
- 2) **Wavelength:** the time in seconds that a pulse reaches its full cycle and starts to repeat itself.
- 3) **Amplitude:** the maximum value and (the negative) minimum value of the pulse



Example: a breathing light .

**Tips:**

1) An easy way to understand the "DELAY" node is: the output is always certain seconds behind the input.



## Software Nodes Card



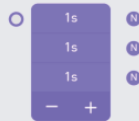
**Nodes Name:** Sequence

**Explanation:** Perform a series of actions according to their time.

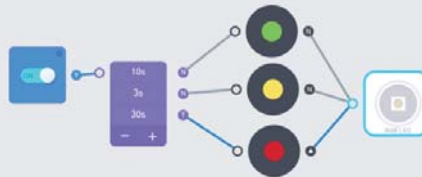
**How it is used:** The “SEQUENCE” node turns on each output one-by-one for a period of time. It can be used for:

- 1) Define an action as a series of moves. Such as shaking heads or waving hands;
- 2) Perform a certain action when a certain condition is met;

**How it works:**



Connect the action to the input of the SEQUENCE node; connect the moves to each of its outputs. Tap on the numbers on the node to set the duration of each output. And use the plus and minus sign to add/remove outputs.



Example: a traffic light that greens for 10 seconds, yellows for 3 seconds, and red for 30 seconds.



Example: when there is a person in front of the device (detected by the PIR sensor), nod head by repeatedly shaking the servo attached to port 1; otherwise, shake head by repeatedly shaking the servo attached to port 2.

## Software Nodes Card



**Nodes Name:** Random

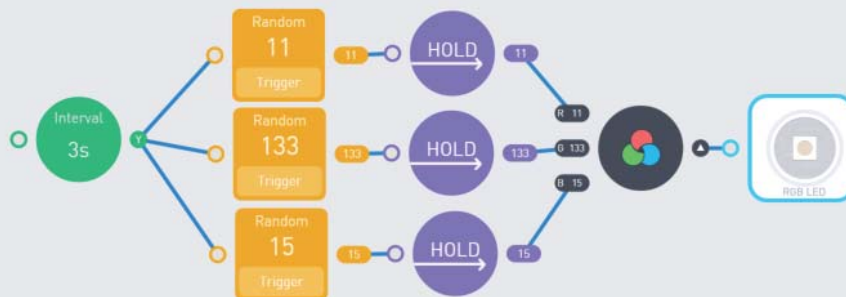
**Explanation:** Make an irregular output every time.

**How it is used:** The RANDOM node can generate a random number. It is used to: - Make a dice, or make an effect that produces a different result every time.

**How it works:**



There are 3 outputs in this node, and the meaning is straightforward: the day of the month(D), the month(M), and the day of the week(W) of today.



Example: a light that changes its color randomly

## Software Nodes Card



**Nodes Name:** Scale

**Explanation:** Map the input from a range to another.

**How it is used:**

- 1) Cases when the given input ranges from 0-100, but the output needs to be 0-255.
- 2) When the input changes too steadily or drastically, you can scale it to the range you want;

**How it works:**



Set two ranges from the config panel. The input will be scaled according to the set range.



Example: make a thermometer. The temperature (0~50) is scaled to the angle of the servo (0~70) that has a needle stick on it that acts as a pointer.



Example: You can use the Scale node to reverse a value. make a light that changes according to the temperature. When it is hot, it turns red; when it is cold, it turns blue. The first SCALE node scales the range of temperature to 0~255, the range of light color. The second SCALE node makes the “blue” value changes to the opposite direction of the red value.

## Software Nodes Card



**Nodes Name:** Filter

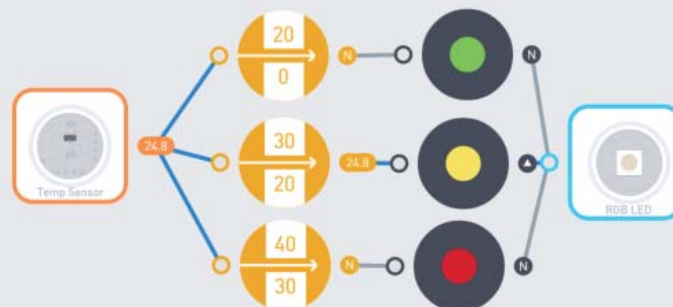
**Explanation:** Only output the input if it falls within the certain range.

**How it is used:** Act differently when the input is in the different range. It makes your program more concise than using a bunch of COMPARE nodes.

**How it works:**



Set a range in the config panel. If the input falls in the range, output the input itself; otherwise, output NO.



Example: when the temperature is 0~20, turn green; when it is 20~30, turn yellow; when it is 30~40, turn red.

## Software Nodes Card



**Nodes Name:** Function

**Explanation:** Pass the input through a mathematical function.

**How it is used:** In case you needs to do some mathematical operation, like rounding a number to its nearest integer, FUNCTION comes to help.

**How it works:**



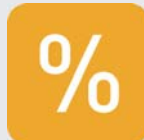
Set the function you want to use in the config panel. The output will always be the input passing through the function you choose.

The settings panel of the function



Example: a knob that can set an integer number from 1 to 10.

## Software Nodes Card



**Nodes Name:** Compute Plus

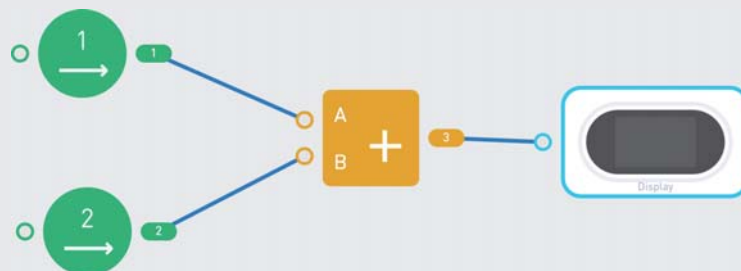
**Explanation:** Make arithmetical operation on two inputs.

**How it is used:** Pick an operator(+, -, \*, /) , make the result variates according to two input sources.

**How it works:**



Set the operator(+, -, \*, /) in the config panel. The result will be .



Example: if you connect A to the upper input, B to the lower input, and the operator is +, then the result will be A+B.

## Software Nodes Card



**Nodes Name:** Compare Plus

**Explanation:** Compare the value of two input sources.

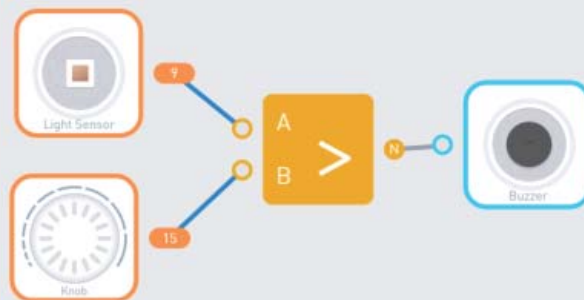
**How it is used:**

- 1) Make a comparison where all operands can change.
- 2) Use the value of a sensor or input device as the threshold of an alarm system.

**How it works:**



Set the operator (>, <, =) in the config panel. Connect two operands to the input. The result will YES if the comparison condition is met. Otherwise it will output NO.



Example: a light-sensor-based alarm device whose sensitivity is controlled by a knob.

## Software Nodes Card



**Nodes Name:** Valve

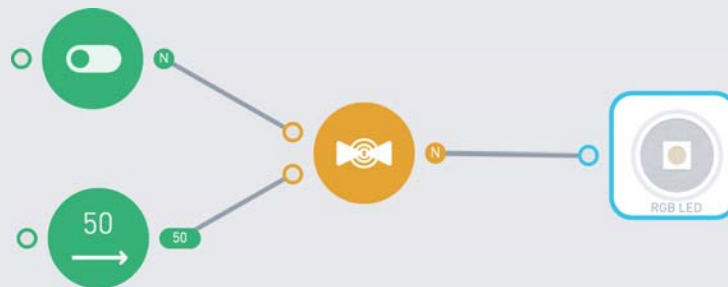
**Explanation:** Output some value if the condition is met.

**How it is used:** “Conditional” nodes like COMPARE only outputs YES or NO; in case you want to output a value other than YES/NO, you will need the VALVE node.

**How it works:**



The upper input takes YES/NO; the lower input takes a value. If the upper input is YES, the lower input is sent to the node's output; otherwise the node will output NO.



Example: While controlling the LED light switch, you can enter the value to control the brightness of the light.