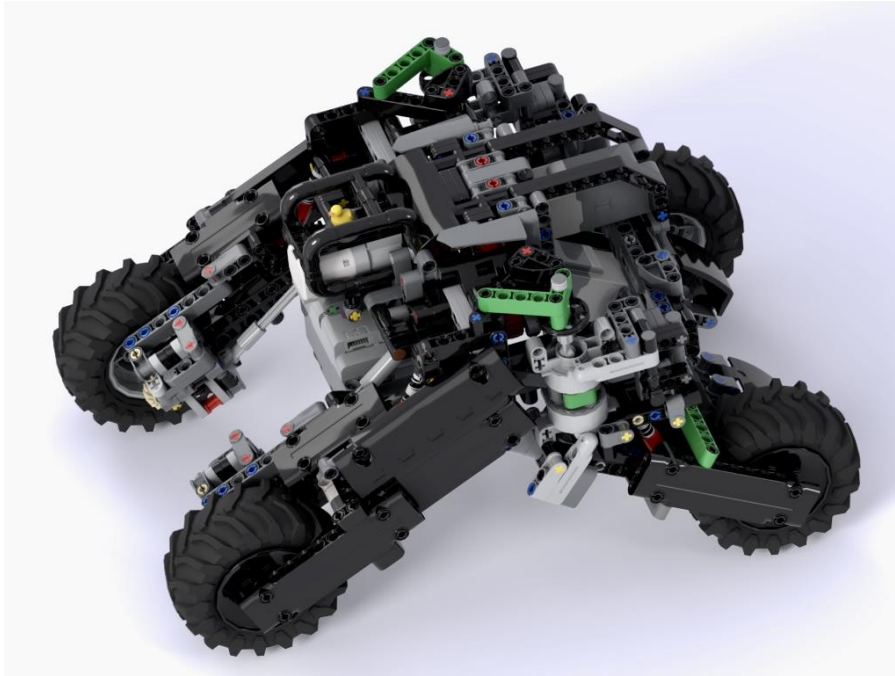


# QUADRUPED 4WD PROGRAM MAKING INSTRUCTION



May 13<sup>th</sup>, 2024

Made by Keisuke Omori

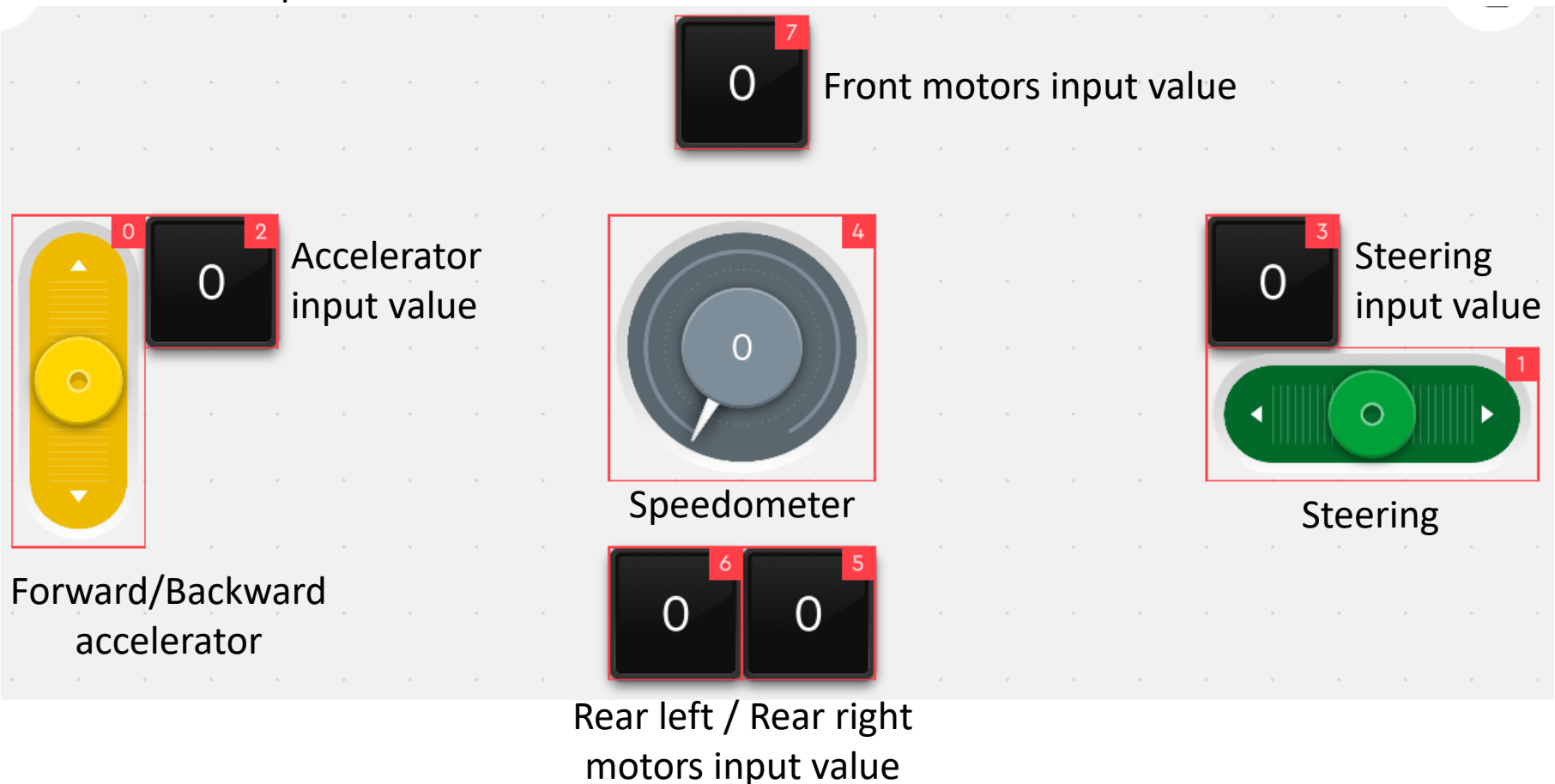
The author's wife has the copyrights.

At the first, please download and setup the Powered UP app in your smart device.  
Imitate the control panel and programs one by one following the order listed below.

## 1 Control Panel

Arrange and number the meters and levers as shown below.

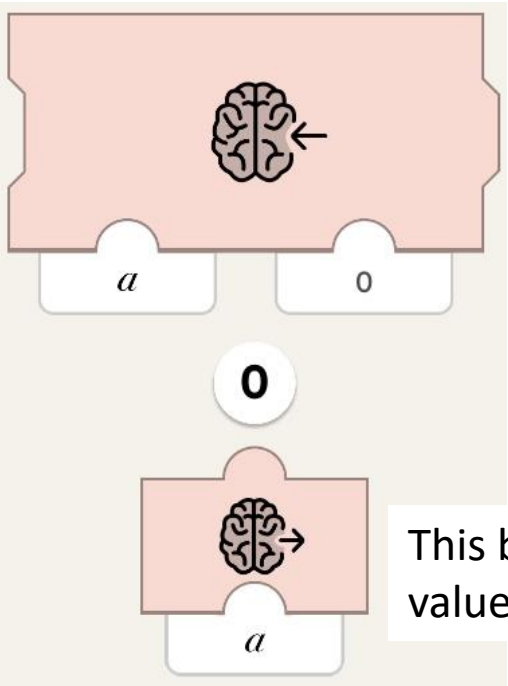
Turn on the option of all levers “Return to 0”.



In these programs, there are some values.  
This MOC uses 7 values as right table.

It doesn't matter if you don't memorize these,  
but I describe these for your understanding.

Alphabet	Usage of value	
a	Motor A (rear right) acceleration	
b	Motor B (rear left) acceleration	
c	Motor C & D (front) acceleration	
d to n	None	
o	Operation status 0 = Not operated 1 = Operated	
p to r	None	
s	Speed value for speedometer	
t, u	None	
v	Lever 0 input value	Temporary memory to know changes in input amount
w	Lever 1 input value	
x to z	None	



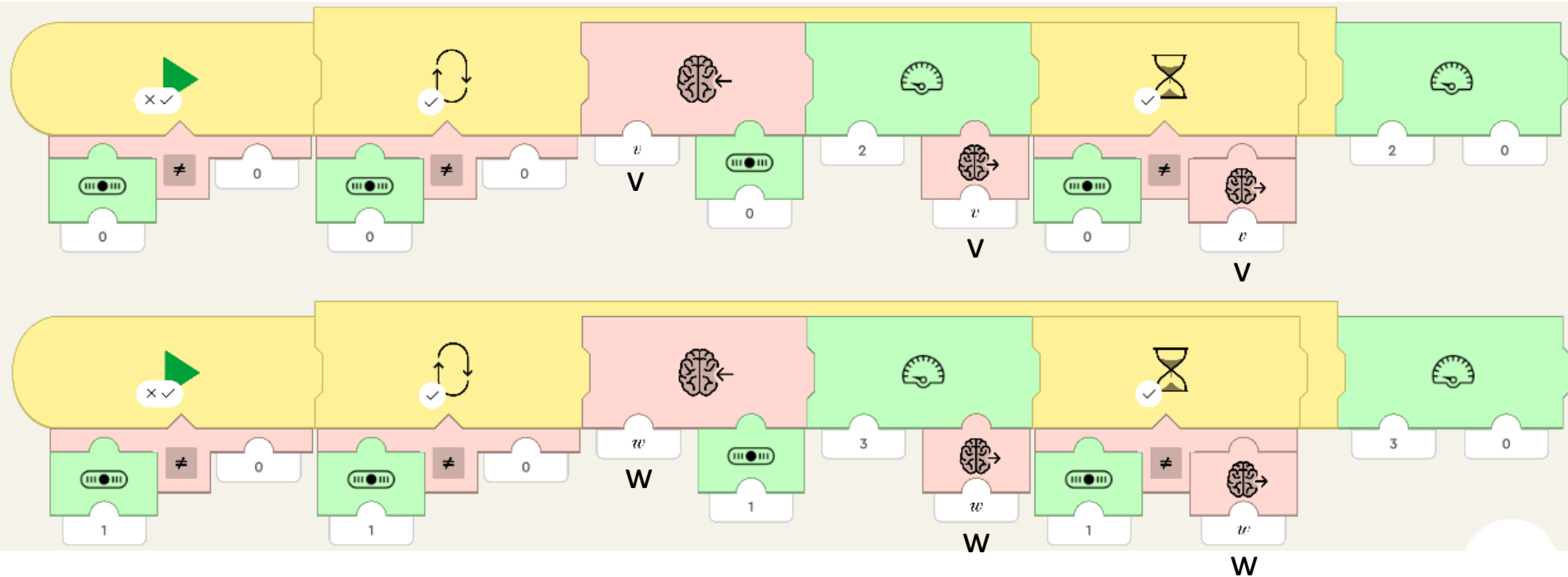
This block sets “a=0”.

This block quotes the value of a. (Now it's 0)

## 2 Programs

Set Program blocks to make program bars one by one. Please be careful because there are many types of program blocks that look very similar. Too small characters are displayed in large size next to them.

### 2-1-1 Display of input values



Each time lever 0 and/or lever 1 are operated and the amount of operation changes, the value is memorized and displayed. The calculation load is reduced by waiting for changes in the amount of lever operation.

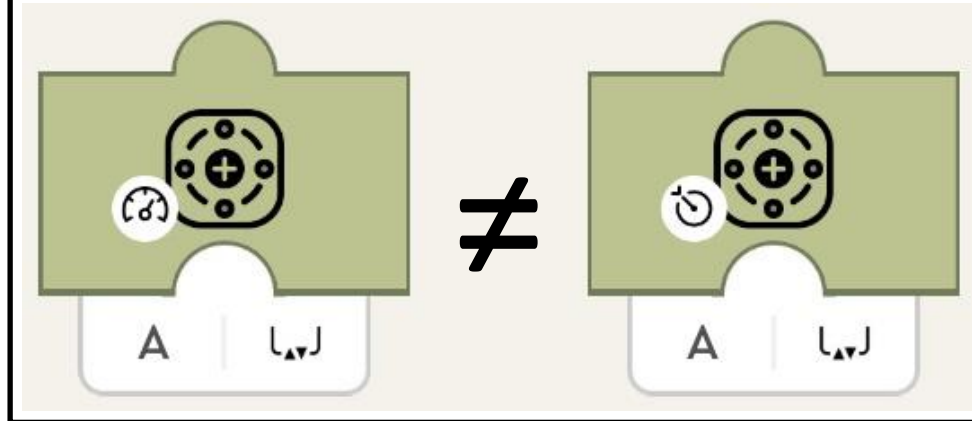
CAUTION! These are different. Color marked.



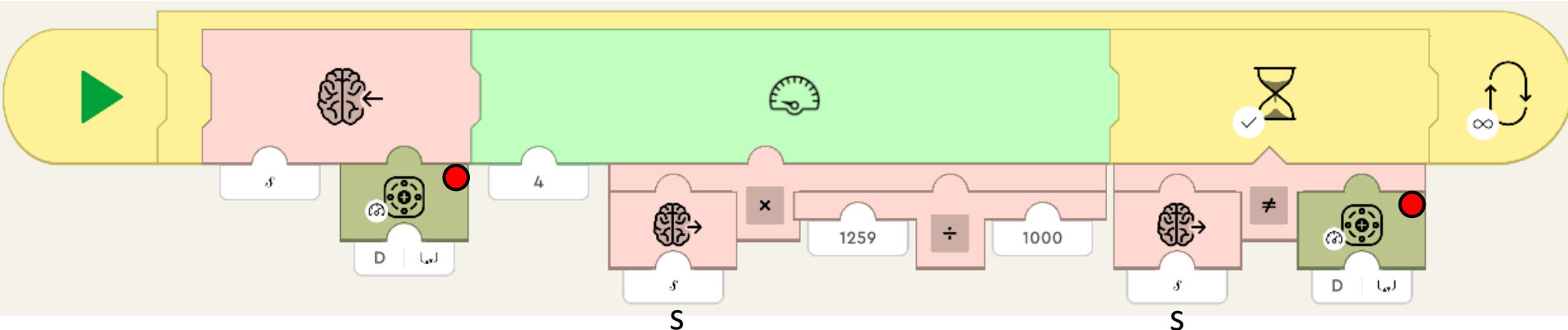
Speed value



Angle value



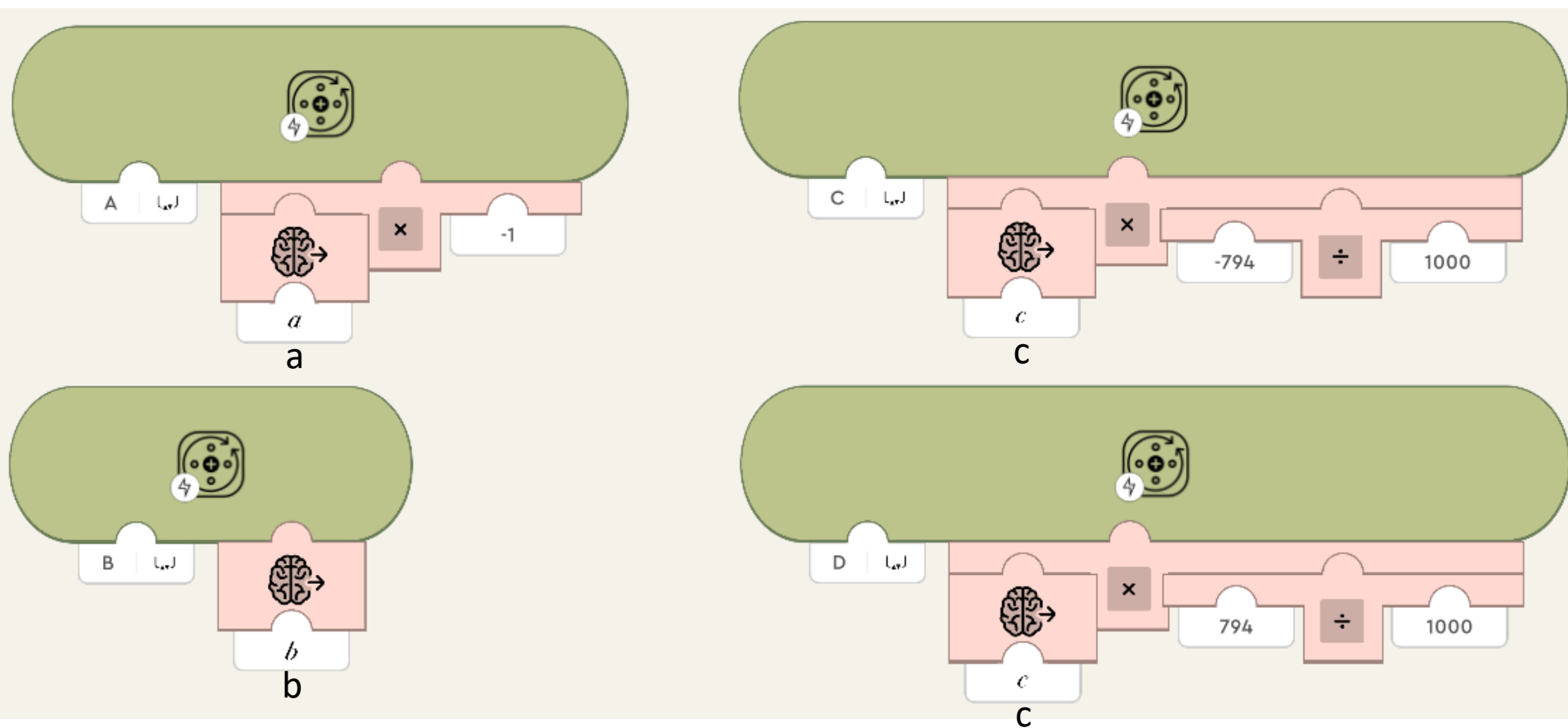
## 2-1-2 Display of speedometer



Update the speedometer display every time the speed changes.

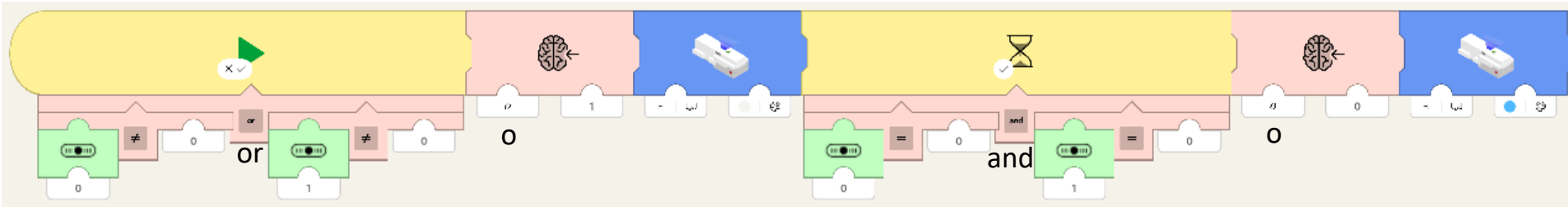
Motor D is required to output only 79.4% at maximum, so the displayed amount is corrected to 1.259 times.

## 2-2 Master motor control



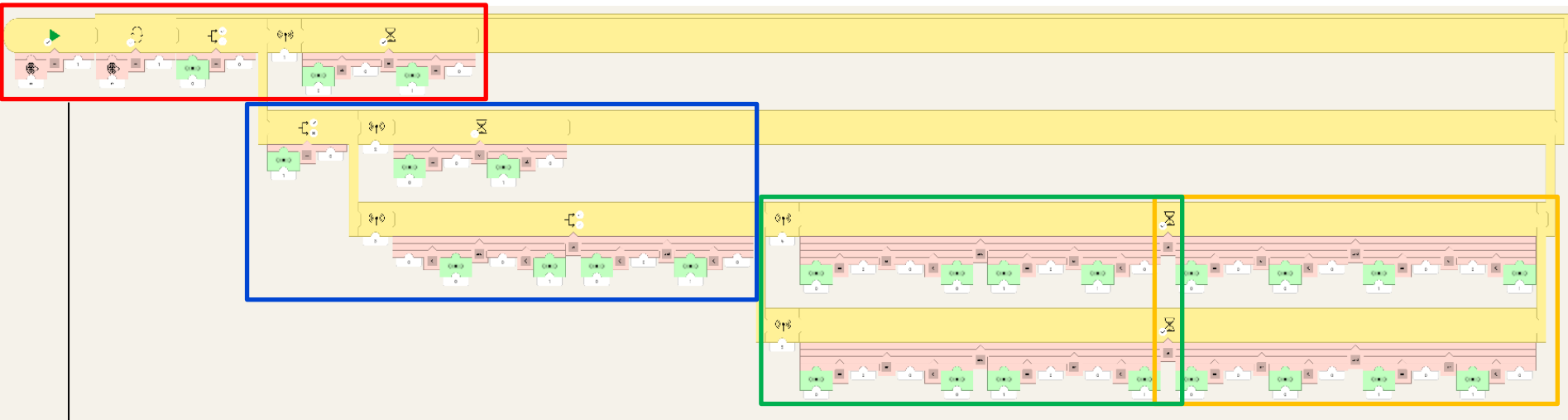
Four motors rotate based on their respective coefficients (value a, b and c). Motors A and C have negative coefficients due to the motor installation positions. Motors C and D, which control the front wheels, have a maximum output of 79.4% to synchronize the rotation speed of the rear wheels due to the gear ratio difference and frictional resistance.

## 2-3 Operation status



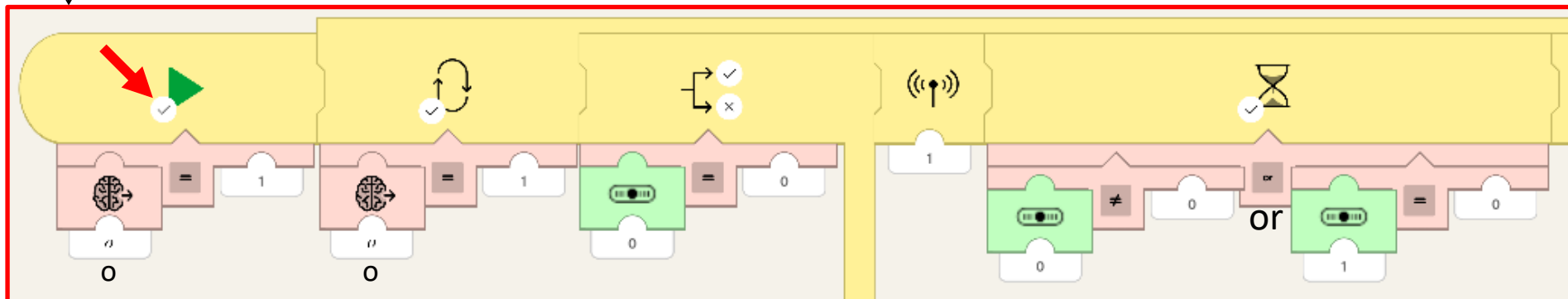
As long as any lever is operated, the operation status becomes 1. During this time, the Technic hub LED will become a colorless headlight. When all levers are not operated, the operation status becomes 0. The LED returns to blue.

## 2-4 Mode selection according to operation



This program is quite large, so please look at the color-coded magnification sections to create one.

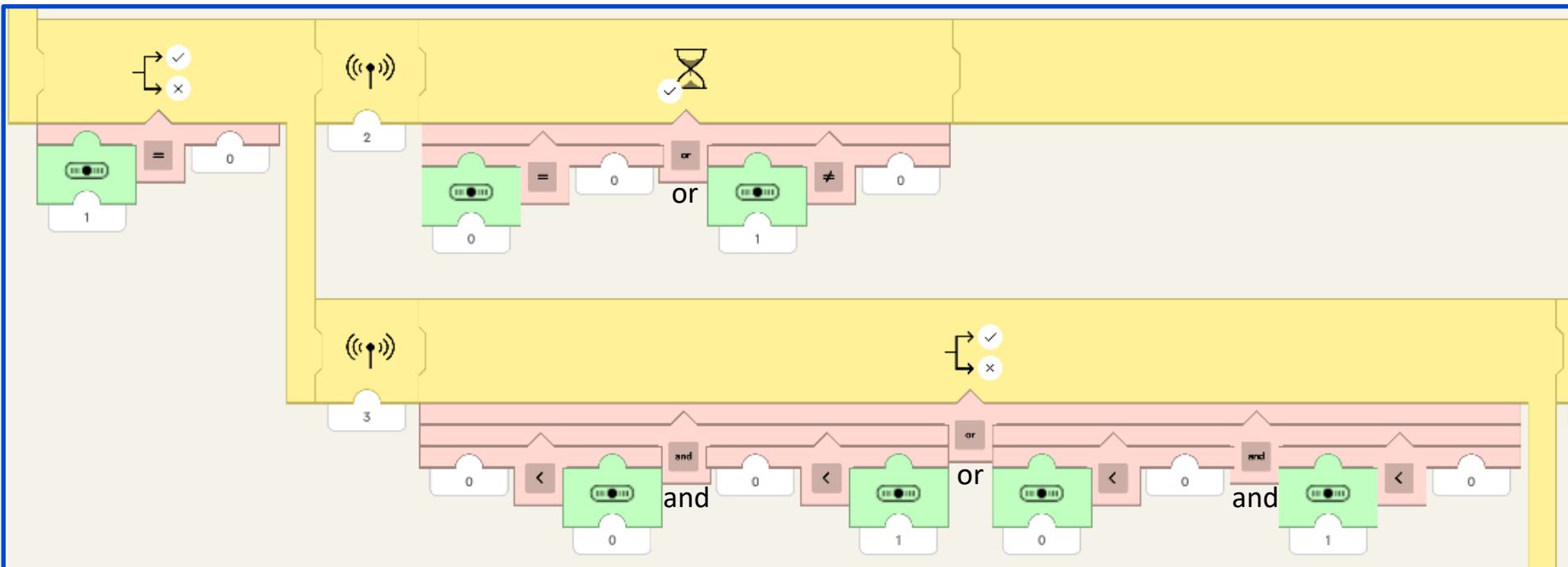
### 2-4-1 Red part



As long as the operation status is 1, the calculation method for values a to c is selected depending on the lever operation status. If the operator is not operating the accelerator (lever 0) and only operating the steering (lever 1), the signal 1 calculation is activated. It waits for the next selection until operator operates the accelerator (lever 0) or stops steering (lever 1).

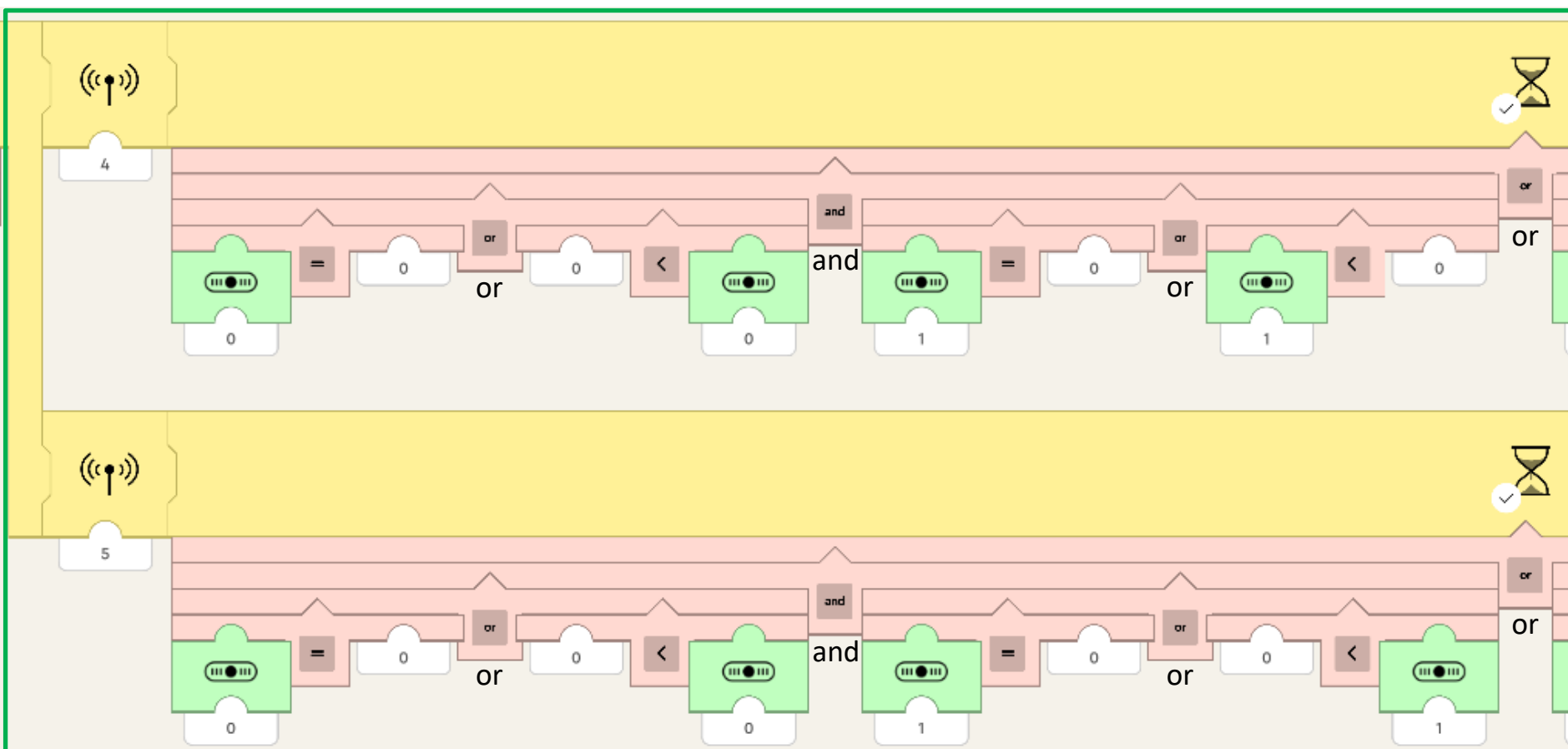


## 2-4-2 Blue part



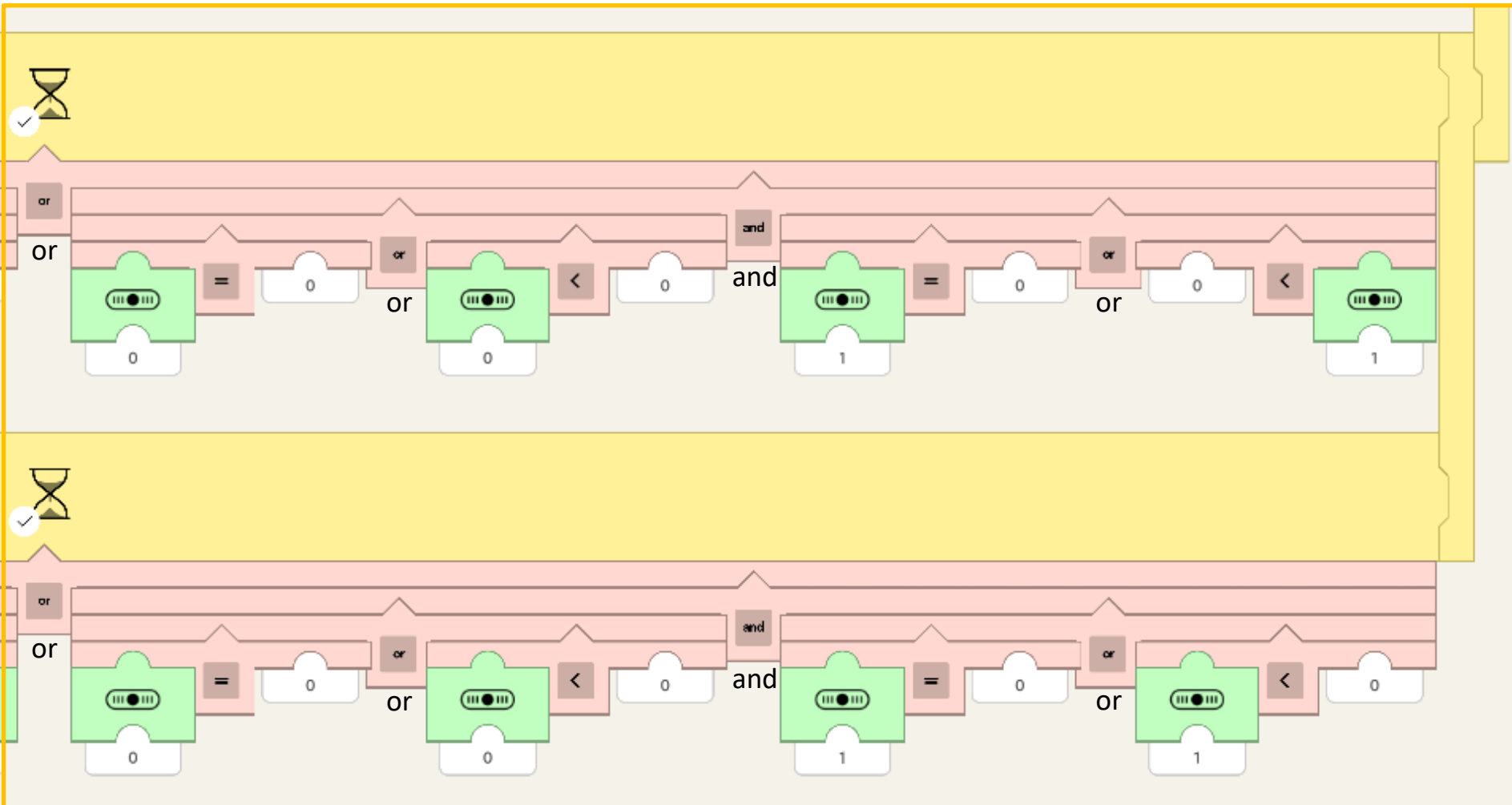
When the operator operates the accelerator (lever 0) but not the steering (lever 1), the signal 2 calculation is activated. When the operator operates both the accelerator (lever 0) and the steering (lever 1), the signal 3 calculation is activated, and then it is determined whether the operation is a right turn forward or a left turn backward.

### 2-4-3 Green part



If the operator operates a right turn forward or a left turn backward, signal 4 calculation is activated. If the operator operates a left turn forward or a right turn backward, signal 5 calculation is activated. The standby conditions are described in next 2-4-4 orange part.

## 2-4-4 Orange part

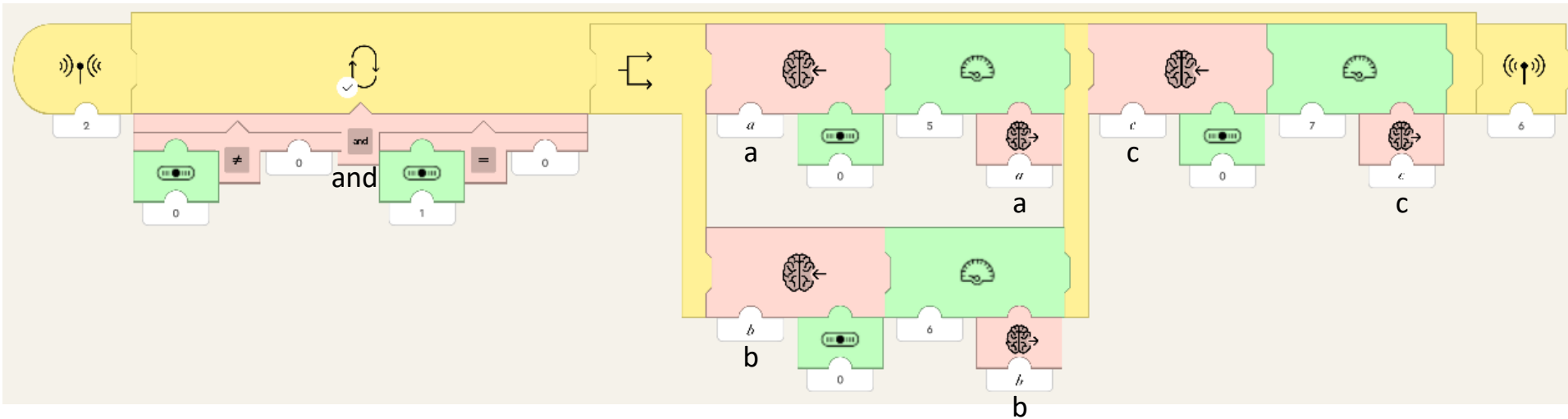


After activating the calculation of signal 4 or 5, the next calculation selection is waited until the operator releases the lever or the lever is operated to a different calculation area.

The diagram shows a complex assembly line with multiple paths and loops. The main path starts with a yellow block (1), followed by a pink block (c), a green block (0), a yellow block (7), and a green block (0). This leads to a large yellow block containing a loop icon. Below this loop, there is a pink block with an equals sign (=) and a green block with a plus sign (+), both labeled 'and'. This is followed by a pink block with a not equal sign (≠) and a green block with a minus sign (-), both labeled '1'. The path then splits into two main branches. The top branch consists of a pink block (a), a green block (x), a pink block (-1), a green block (5), and a pink block (a). The bottom branch consists of a pink block (b), a green block (1), a pink block (6), and a green block (b). Both branches converge into a final yellow block (6).

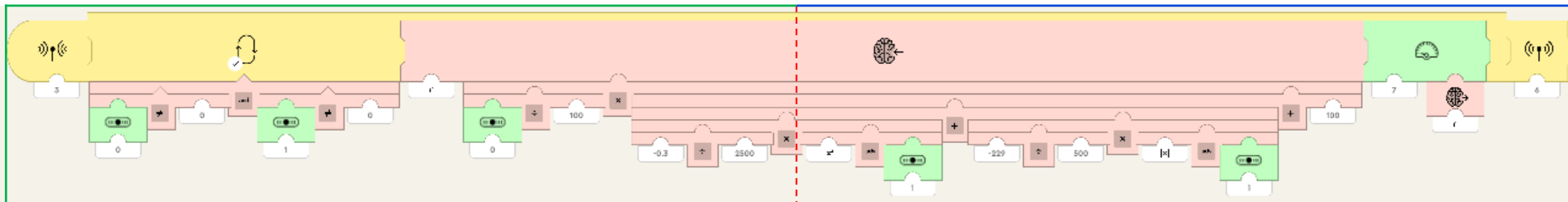
This is a program for on-the-spot steering, which twists the vehicle by moving the rear wheels in response to the amount of steering (lever 1) operation without moving the front wheels. Displays the amount of operation when moving the rear wheels. When the operator operates the accelerator (lever 0) or stops steering (lever 1), signal 6 is activated.

## 2-6 Signal 2, Straight movement

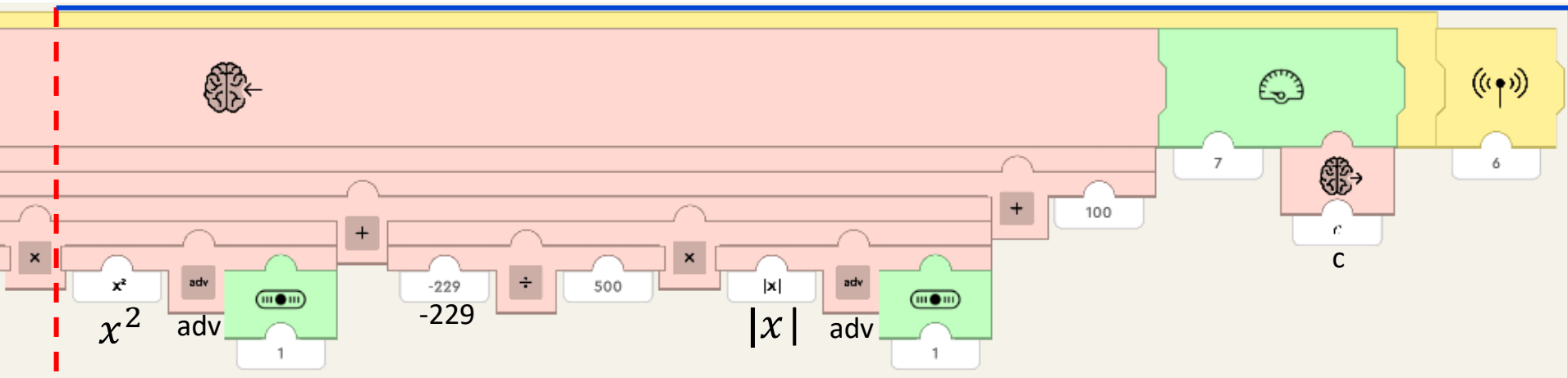
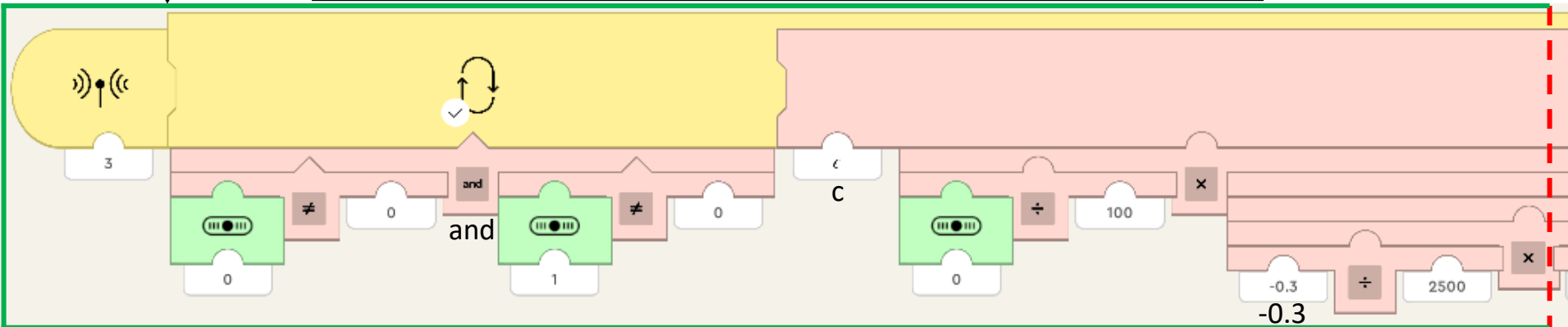


This is a program for straight movement, as long as the operator is operating the accelerator (lever 0) and not operating the steering (lever 1), all motors will output power proportional to the amount of accelerator operation, and the vehicle will move straight forward or straight backward. Displays the amount of operation when moving the front and rear wheels. When the operator operates the steering (lever 1) or stops the accelerator (lever 0), signal 6 is activated.

## 2-7 Signal 3, front wheel control



This program is quite large, so please look at the enlarged version in two parts below.

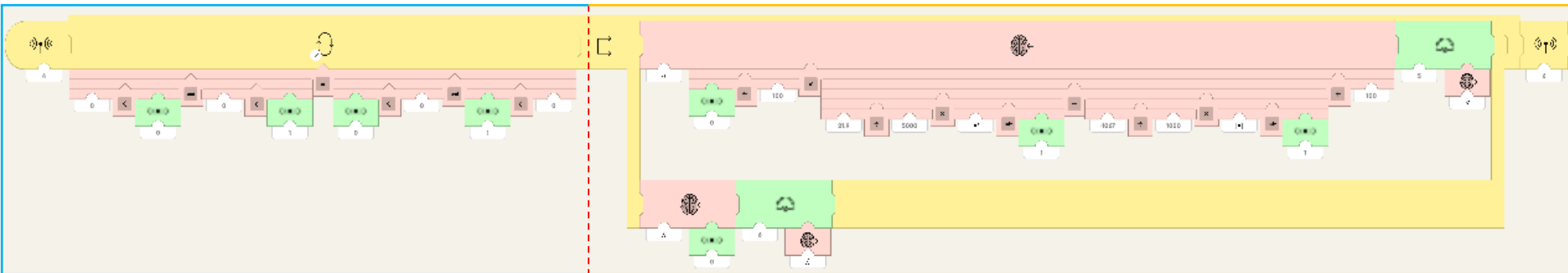


The values for the front wheels are determined as follows, regardless of whether the vehicle turning left or right. This calculation and display is repeated as long as the lever continues to be operated.

$$c = (0.01v) \times [(-0.00012w^2) + (-0.458|w|) + 100]$$

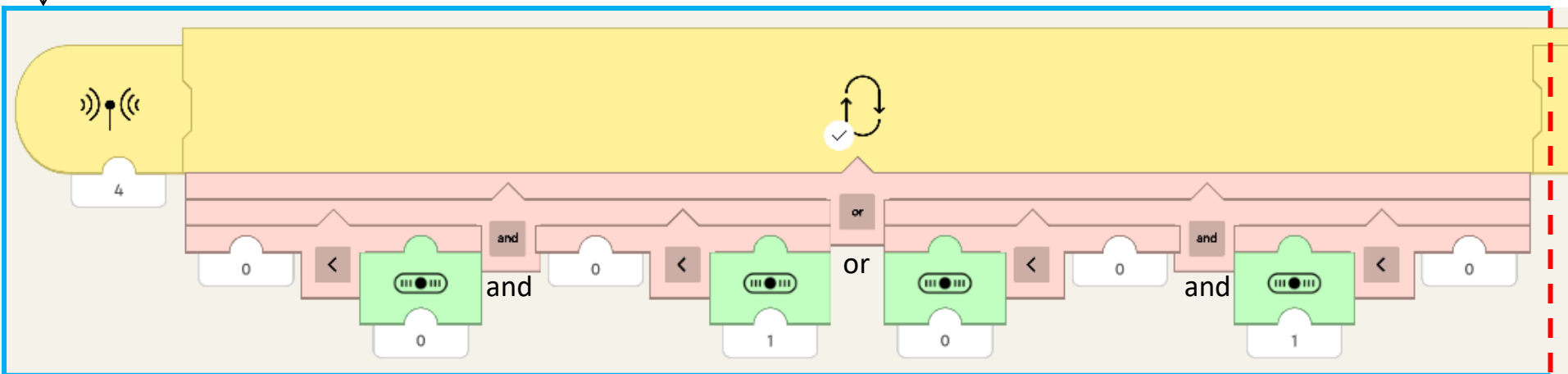
When the operator stops operating at least one of the levers, signal 6 is activated.

## 2-8 Signal 4, rear motor control 1



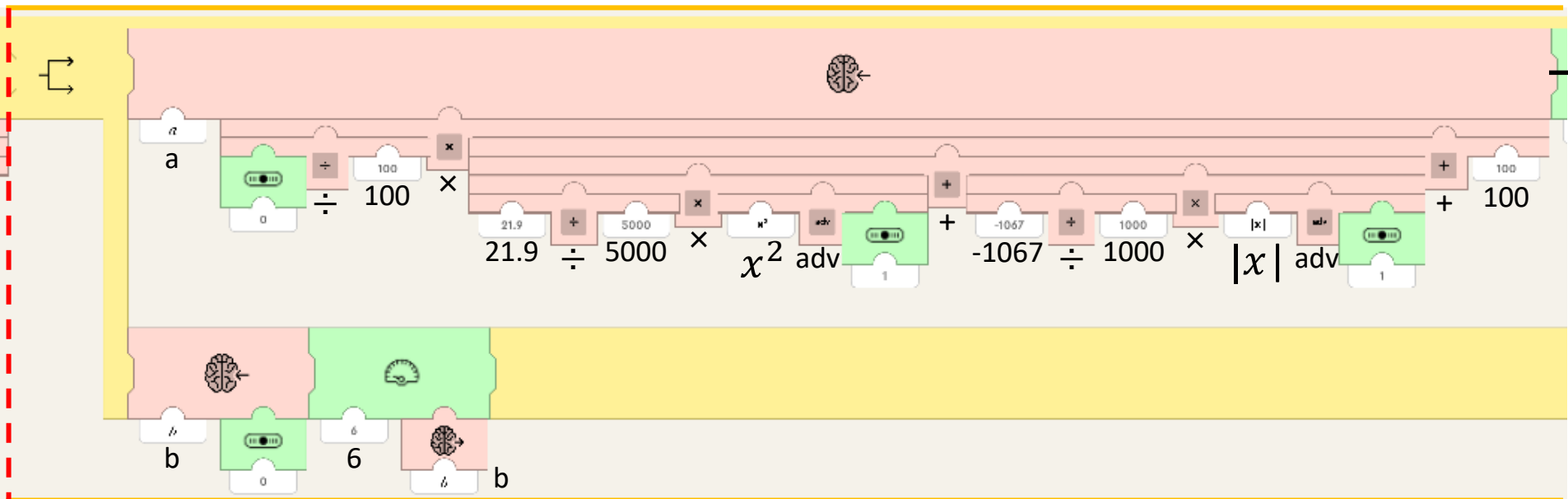
This program is quite large, so please look at the enlarged version following.

### 2-8-1 Light blue part



This calculation is repeated as long as the operator continues to make a right turn forward or a left turn backward.

## 2-8-2 Orange part



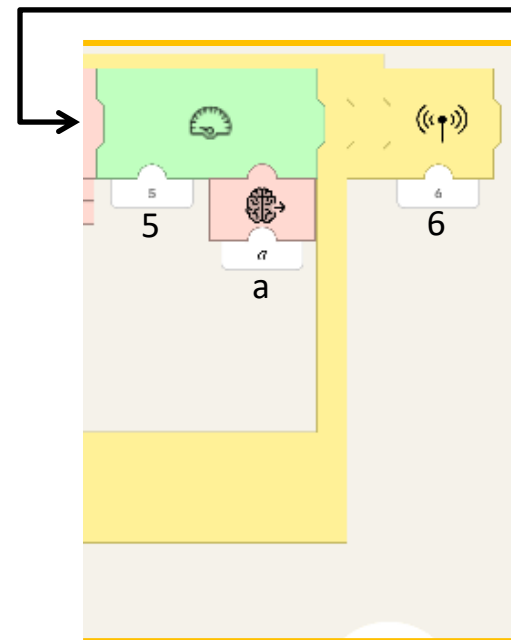
The calculations to repeat are as follows:

$$a = (0.01v) \times [(0.00438w^2) + (-1.067|w|) + 100]$$

$$b = v$$

Value a and b are displayed.

When the operator stops operating or starts operating another motion, signal 6 is activated.

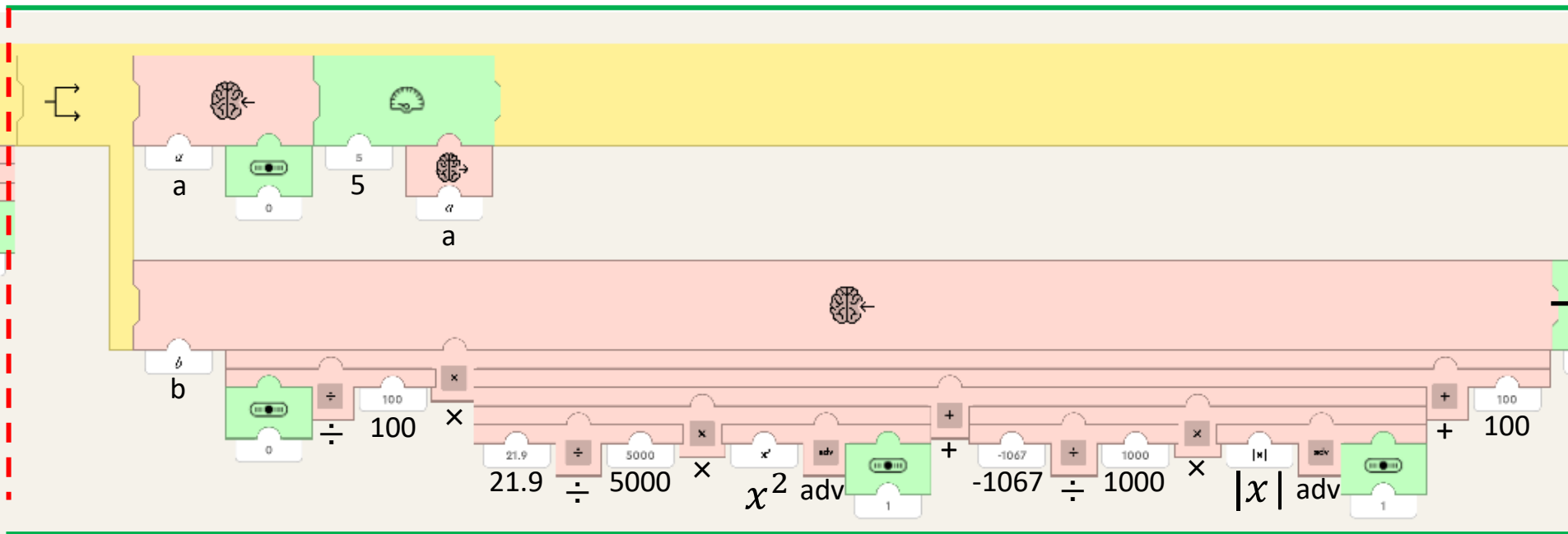




This program is also quite large, so please look at the enlarged version following.

This calculation is repeated as long as the operator continues to make a left turn forward or a right turn backward.

## 2-9-2 Green part



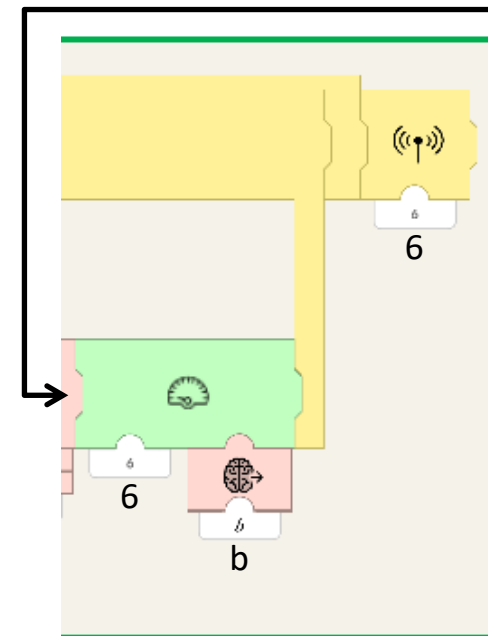
The calculations to repeat are as follows:

$$a = v$$

$$b = (0.01v) \times [(0.00438w^2) + (-1.067|w|) + 100]$$

Value a and b are displayed.

When the operator stops operating or starts operating another motion, signal 6 is activated.

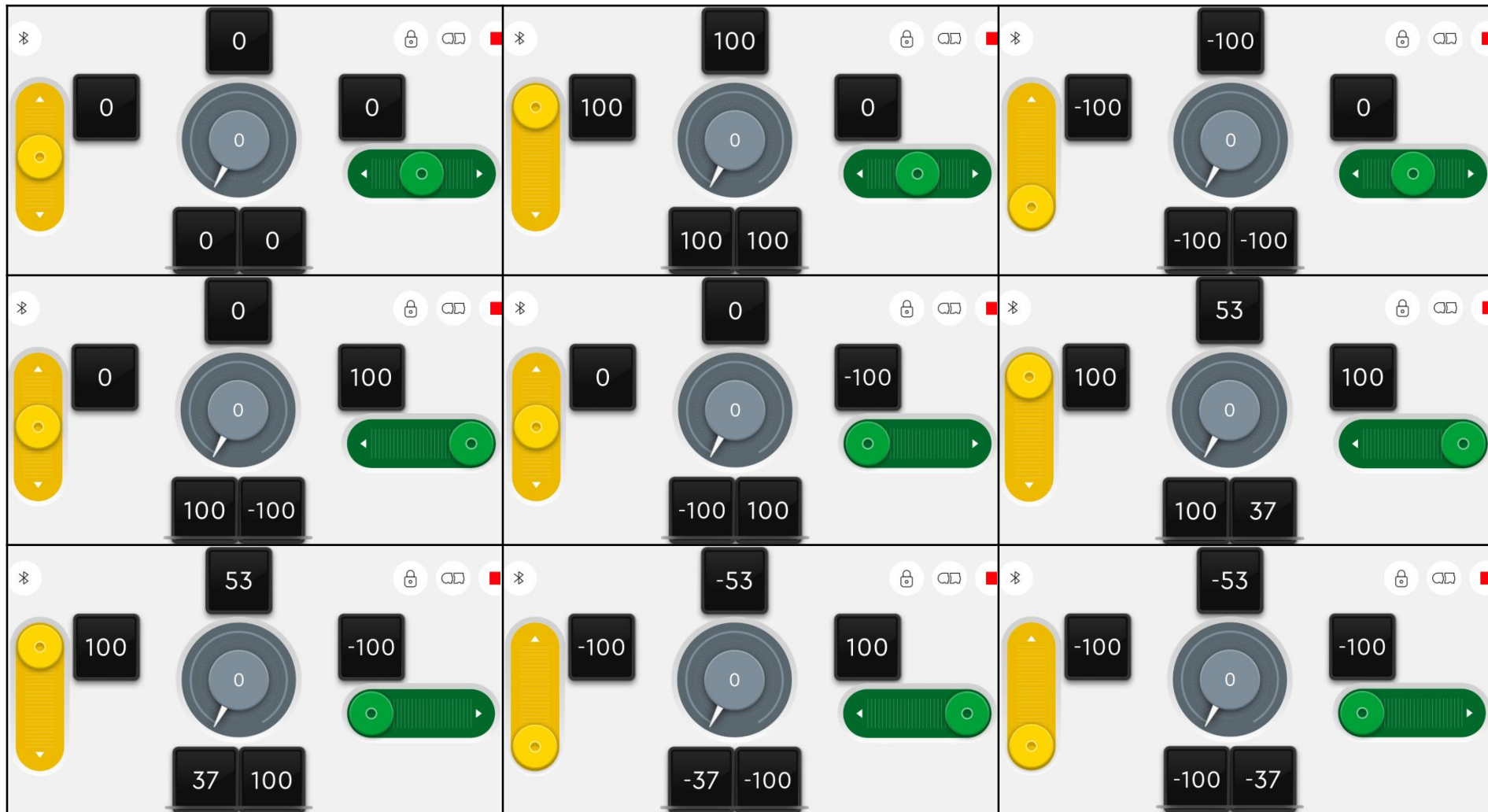


The diagram illustrates a cognitive architecture within a yellow environment. On the left, a robot with a single eye and antennae is positioned near a pink structure. This structure contains a logic gate labeled 'and' with two inputs, each preceded by an equals sign and a white box containing the number 0. Below the inputs are two green boxes, each containing a sensor icon (a circle with a vertical line and a dot) and a white box with the number 0. To the right of the pink structure is a green structure with a sensor icon and a white box with the number 1. Further right, a pink structure contains a logic gate labeled 'and' with two inputs, each preceded by an equals sign and a white box containing the number 0. Below the inputs are two green boxes, each containing a sensor icon and a white box with the number 0. On the far right, a pink structure contains a logic gate labeled 'and' with two inputs, each preceded by an equals sign and a white box containing the number 0. Below the inputs are two green boxes, each containing a sensor icon and a white box with the number 0. The entire scene is set within a yellow environment with a black floor area at the bottom.

The end of repeating signals 1 to 5 means that another operation continues or the levers are released and the calculation cycle ends. There is no problem if another operation continues, but when the operation ends, non-zero noise may remain in value a, b or c, and the motor may not stop. When the levers are no longer operated, reset values a, b and c.

## 2-11 Offline check

Before connecting to the Technic hub, operate the levers offline and check that the values are correct.  
If the lever operations and values do not look like the screenshots below, there is an error in the program, so correct it.



If everything is OK, connect it to Technic hub (vehicle) and try running it.

# Congratulations on completion!

The author is happy if your Technic playing life got exciting.

Please upload the photo of your build.

Let me know whatever you thought on this MOC.

Thank you!

Keisuke Omori