Program Descriptions

GyroBoy

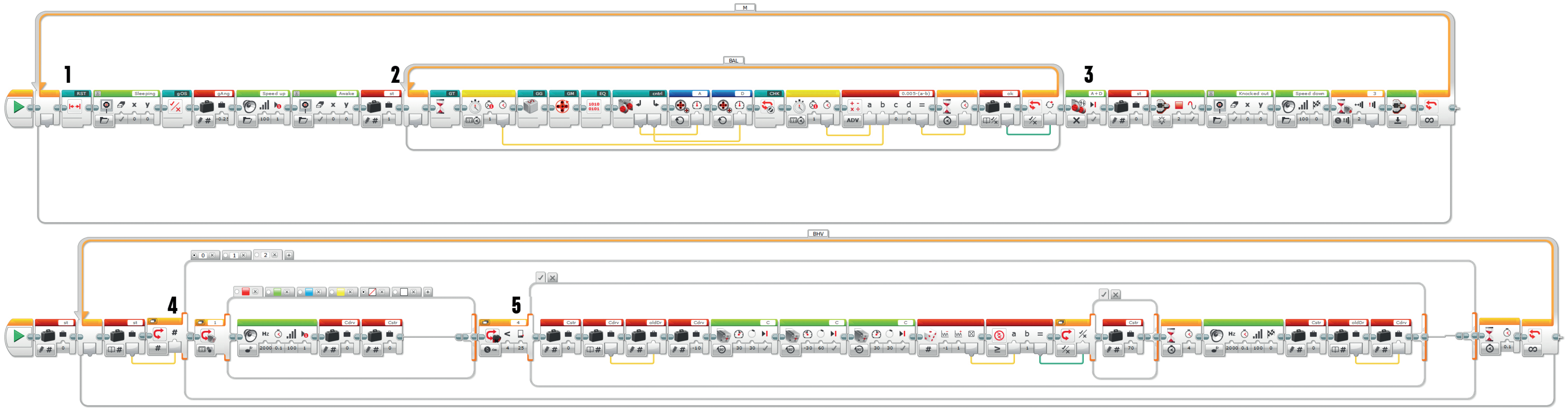
Overview

Gyroboy runs two parallel program strings. The first string, loop M, handles the data collection and balancing equations. These are all tuned to the robot and should not be changed without advanced knowledge. The second loop, BHV, handles the behavior of the robot.
It allows basic control and sensor feedback. By changing the variables Cdrv and Cstr, you can make the robot do what you want.

1 Loop M begins with My Block RST, this will reset all motors and sensors. Myblock gOS checks to see how still the robot is. Once complete, the robot sets up its launch sequence. On Display graphics show an expression for the robot.

2 Loop BAL takes all of the data from the Gyro Sensor and Motor Sensors to process them to balance. Time is also used in the calculations. Timer 1 is used to calculate the time it takes to run the equation, and regulates the time between getting the data and running the balancing equation. The time needs to be regulated in order to have a more stable robot.
The Unregulated Motor Blocks for Motor A and D are used to make sure that only the balancing equation regulates the movement, not any other internal calculations. Myblock CHK is there to check if the robot has fallen, and ends the loop with Variable ok if it does. The My Blocks are shown as they are to improve readability and possible customization.

3 Loop M ends by stopping the motors and behaviors. A change in graphics, sound and brick light display indicate the fallen status and is ready to reset. It should give time for the user to put the robot back on the stand, then by pressing the touch sensor, the robot can start again.



4 Loop BHV controls the behavior of the robot.
State Variable S is wired to A switch in Numeric Mode with 3 options. If Variable S is 0, the Variables CDrv and CStr are set to 0 and this is the idle state of the robot. If S is 1, the robot does a launch sequence. Variable CDrv is set to 40 for 4 seconds, then back to zero then Variable S is changed to 2 in order to begin the movement state.

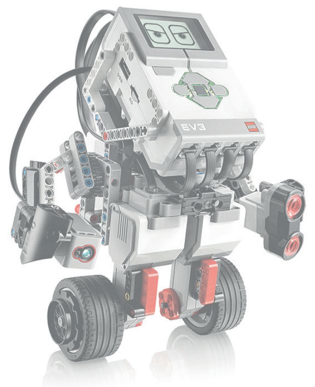
5 When Variable S is 2, it is the main operating and interactive state of the robot.
The Color Sensor is checked and for each available color, there is a different value for variables Cstr and Cdrv. The next Switch in Ultrasonic Mode checks to see if an object is in front of the robot. If there is, the robot stops and saves its last driving condition, then prepares to turn away by moving slightly back, and waving its arms. The robot then turns randomly left or right for a few seconds and goes back to its previous driving state.

www.LEGOeducation.com

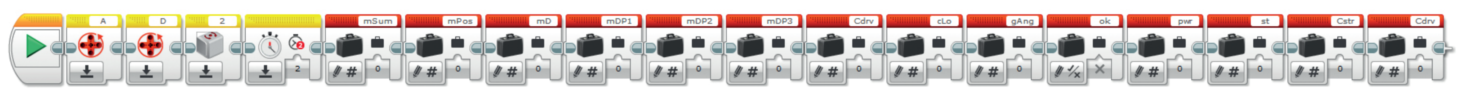
LEGO mindstorms education EV3

Program Descriptions

GyroBoy

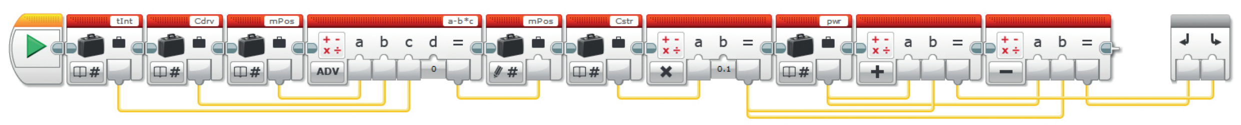


RST



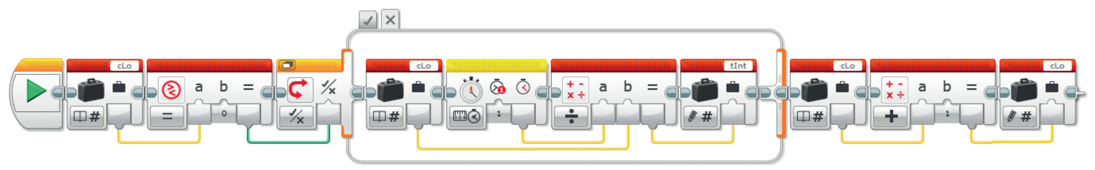
My Block RST resets all motors, sensors, timers and variables that are needed in this program.

Ctrl



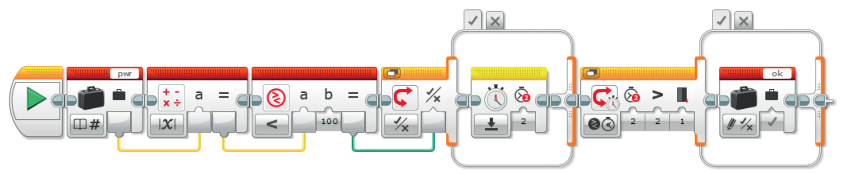
My Block Ctrl uses two variables to control the robot. Cdrv is again used to create a target motor position. The Variable Cdrv is multiplied by tInt and subtracted from the previous mPos to create the target. Variable Cstr is multplied by .1 and, when added to the power, Variable pwr will wire to the left drive motor. By subtracting the wire will go to the right drive motor.

GT



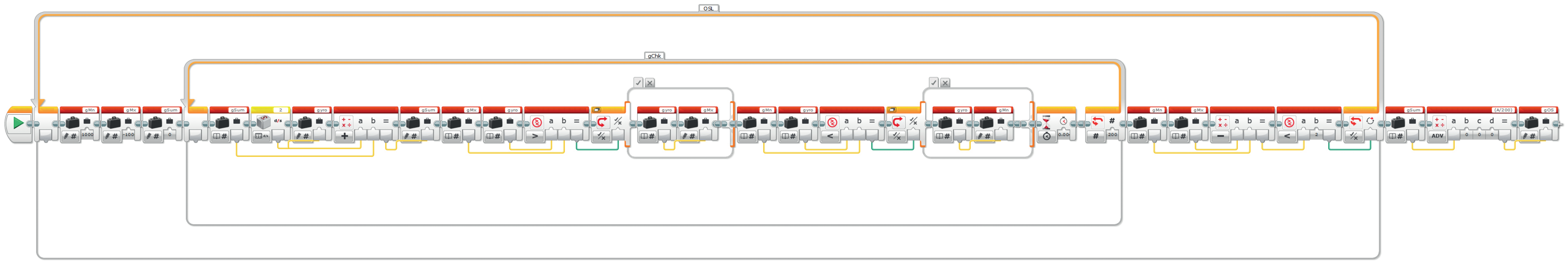
My Block GT calculates a time interval based on a timer and loop count. Timer 1 is divided by Variable cLo, which represents the loop count and wired to the Variable tInt. One is added to Variable cLo after the calculation.

Chk



My Block Chk checks if the robot has fallen. If the power is 100 for more than 2 seconds, the Variable ok will end the balancing loop and the robot can be placed on the stand again.

gOS



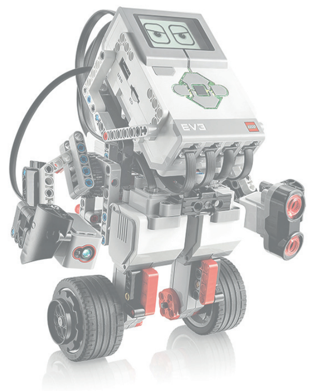
My Block gOS makes an offset value of the Gyro Sensor in its steady position. If the gyro is slightly drifting, this will keep the value regulated as the robot moves. The offset is also dynamically calculated while the robot is running. To calculate this value the gyro value is read and added together 250 times. In each addition, the value is checked at maximum and minimum values to make sure the robot is steady. After Loop gChk ends, the difference between gMax and gMin is checked, if the value is less than 2, the calculation will continue, otherwise the robot will check the gyro again. The average of the gSum is taken by the Math Block dividing by 250, and then this is wired to the Variable gOS.

LEGO education

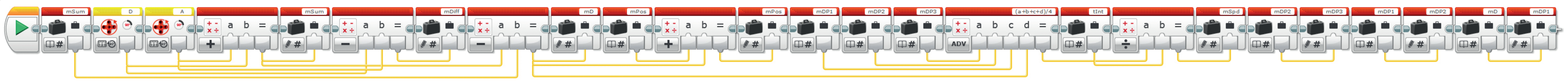
LEGO mindstorms education EV3

Program Descriptions

GyroBoy

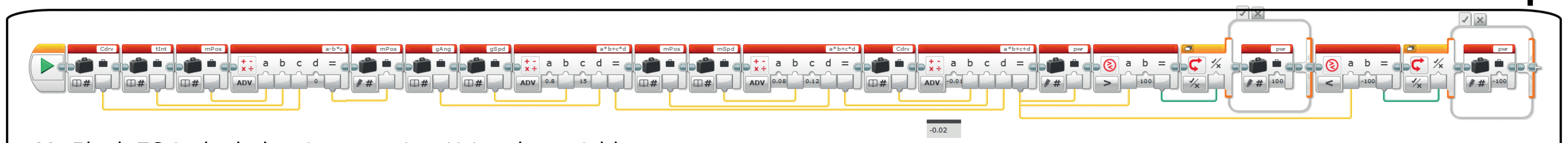


GM



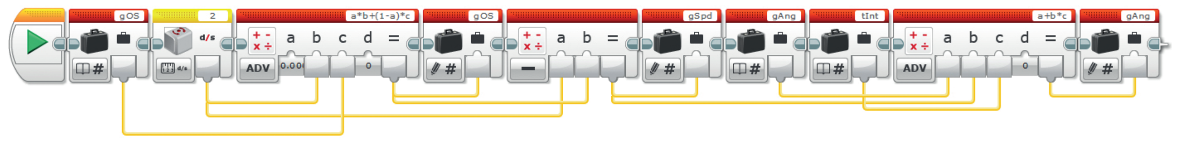
My Block GM gets information from the motors. First, the motor position is calculated by adding the degree values of Motor A and D and subtracting that value from a previous calculation. Variable mD is the initial difference. The difference is added to an old mPos and makes the new mPos. Motor speed, Variable mSpd, is created by taking an average of 4 motor differences and dividing by Variable tInt. The last blocks move the motor differences to other variables in succession.

EQ



My Block EQ is the balancing equation. Using the variables mPos, mSpd, gAng, gSpd and Cdrv, the Variable pwr is created. First create a target mPos using Variable Cdrv. Next, multiply gAng with .8 and gSpd with 15. The same is done with Variables mPos, with .12 and mSpd with .08. These multipliers give weight to the variables that are added together in a Math Block. All are finally added together with Cdrv and multiplier -.02 to create the pwr Variable. Lastly, Variable pwr is checked if it is greater than 100 or less than -100 and sets the value to the maximum or minimum if needed.

GG



My Block GG gets information from the Gyro Sensor. Variable gOS, from the My Block gOS, is used to create a dynamic offset. The Gyro Sensor, reading degrees/seconds is used to create a new offset. The difference between the gyro reading and Variable gOS creates Variable gSpd. Variable gAng is used by multiplying Variables gSpd and tInt. This way is faster than using the direct readings from the gyro and compensates for drifting in the gyro as it happens on the robot.