FLL program

Developement documentation

2022-2023

TABLE OF CONTENTS

1. DATAS

The Robot's Name: MRGT

Development Team's Name: Mobilis Bits MRGT

Team Members: Angster Barnabás, Angster Zsófia, Erdélyi Balázs, Nagy Zsombor Péter, Pintér Dániel, Pintér Gábor, Törös Gergő,

Coaches: Unger-Trescsik Hanna Margit, Boros Norbert**,** Kalocsa Ádám

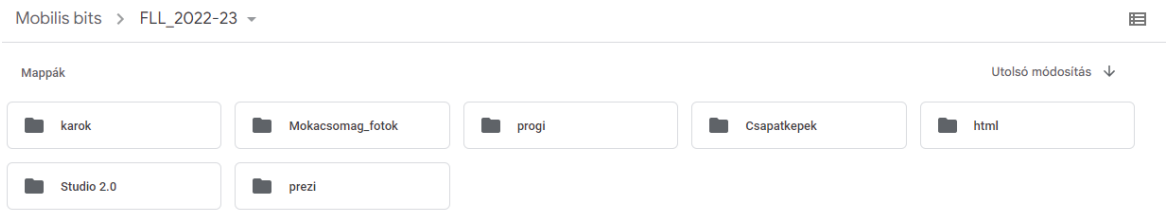
Start of Development: 2021.08.08.

Place of Development: Győr, Mobilis Science Center, Talent Lab

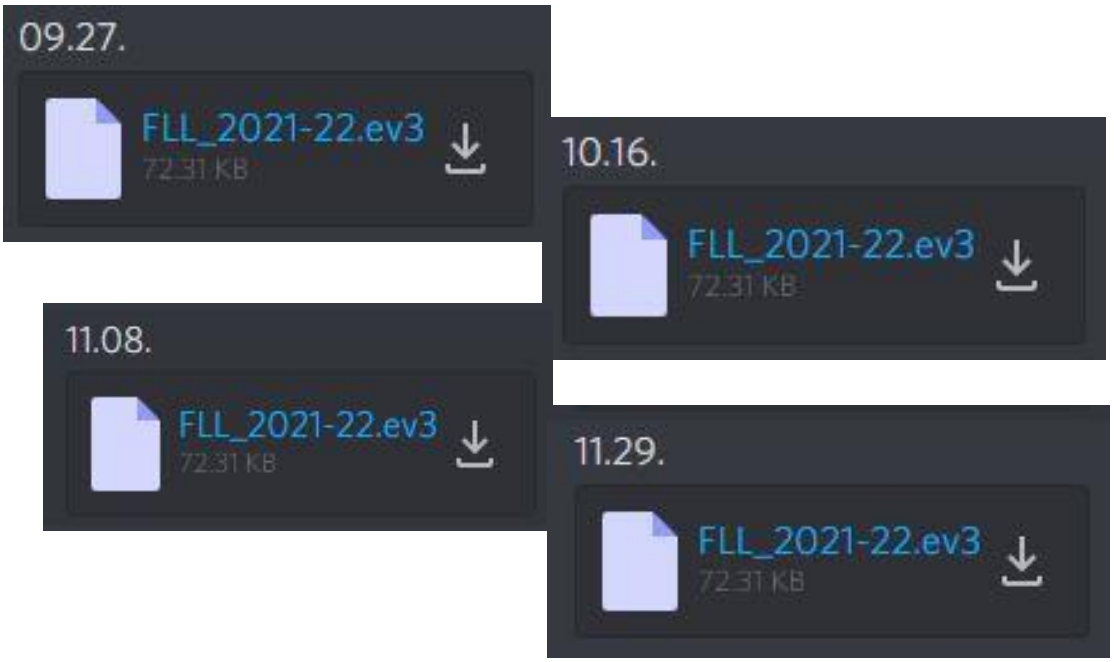
2. SAVE

For safety reasons, we made copies of the programs after every day of research to an external flash drive. We also uploaded the files to a cloud storage using Google Drive. In the last three years we also started using GitHub to store out progress more efficiently.

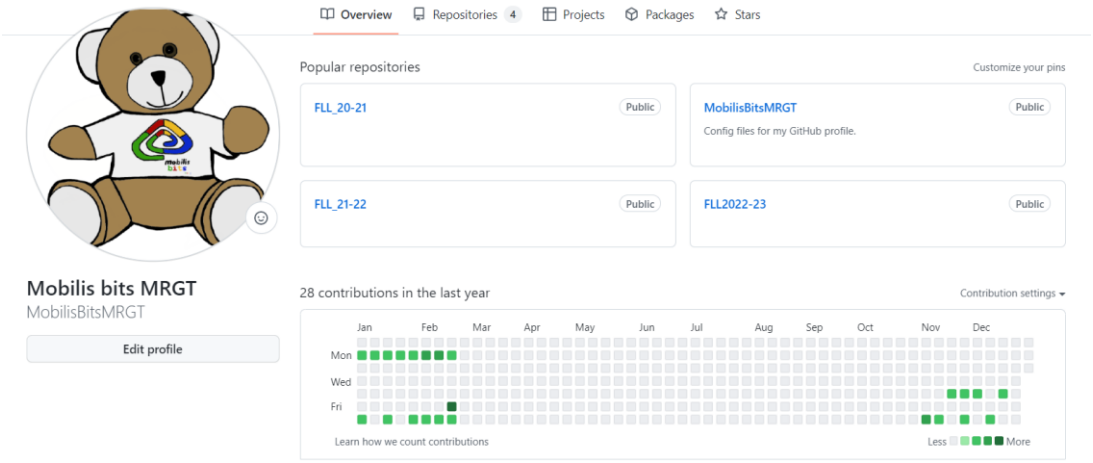
2.1. Google Drive



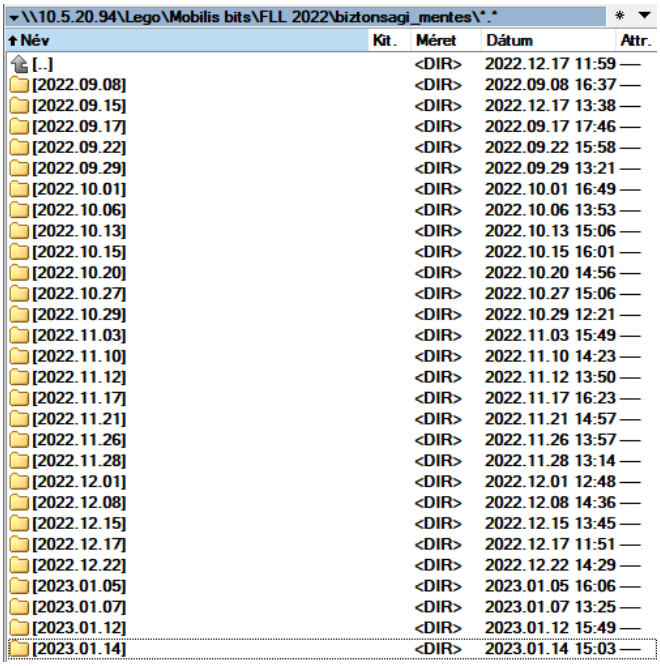
2.2. Discord



2.3. GitHub

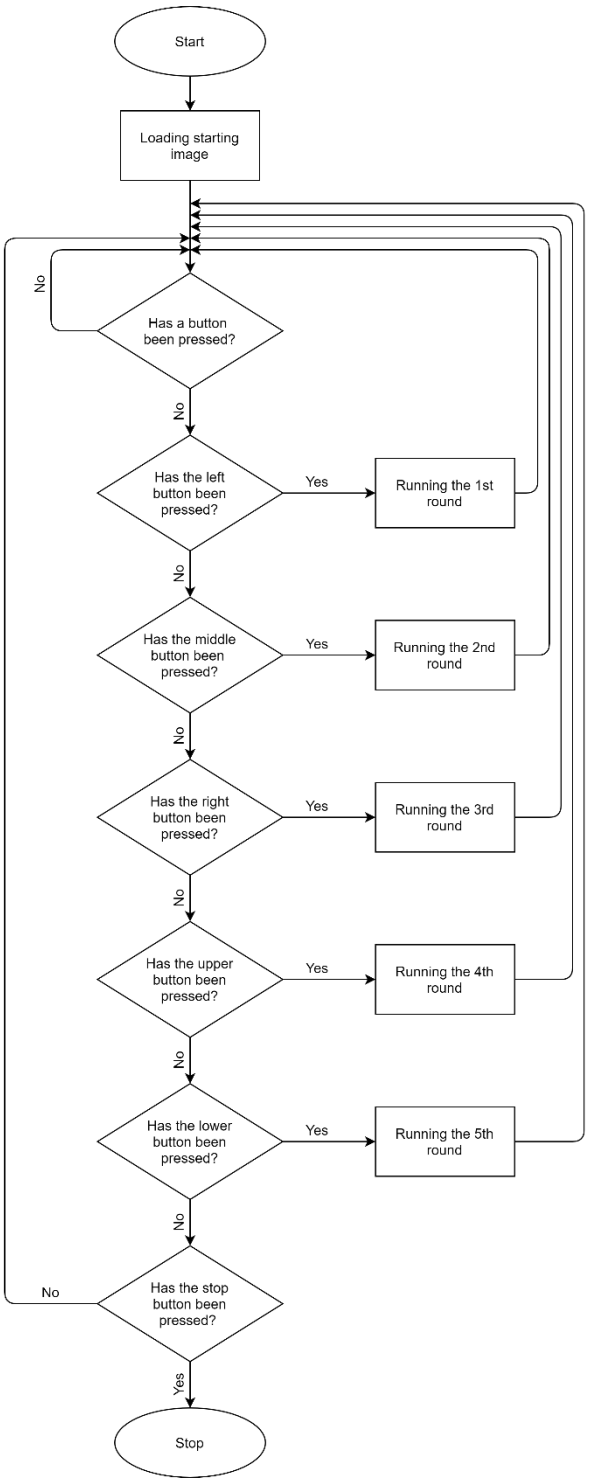


2.4. Network/Pendrive

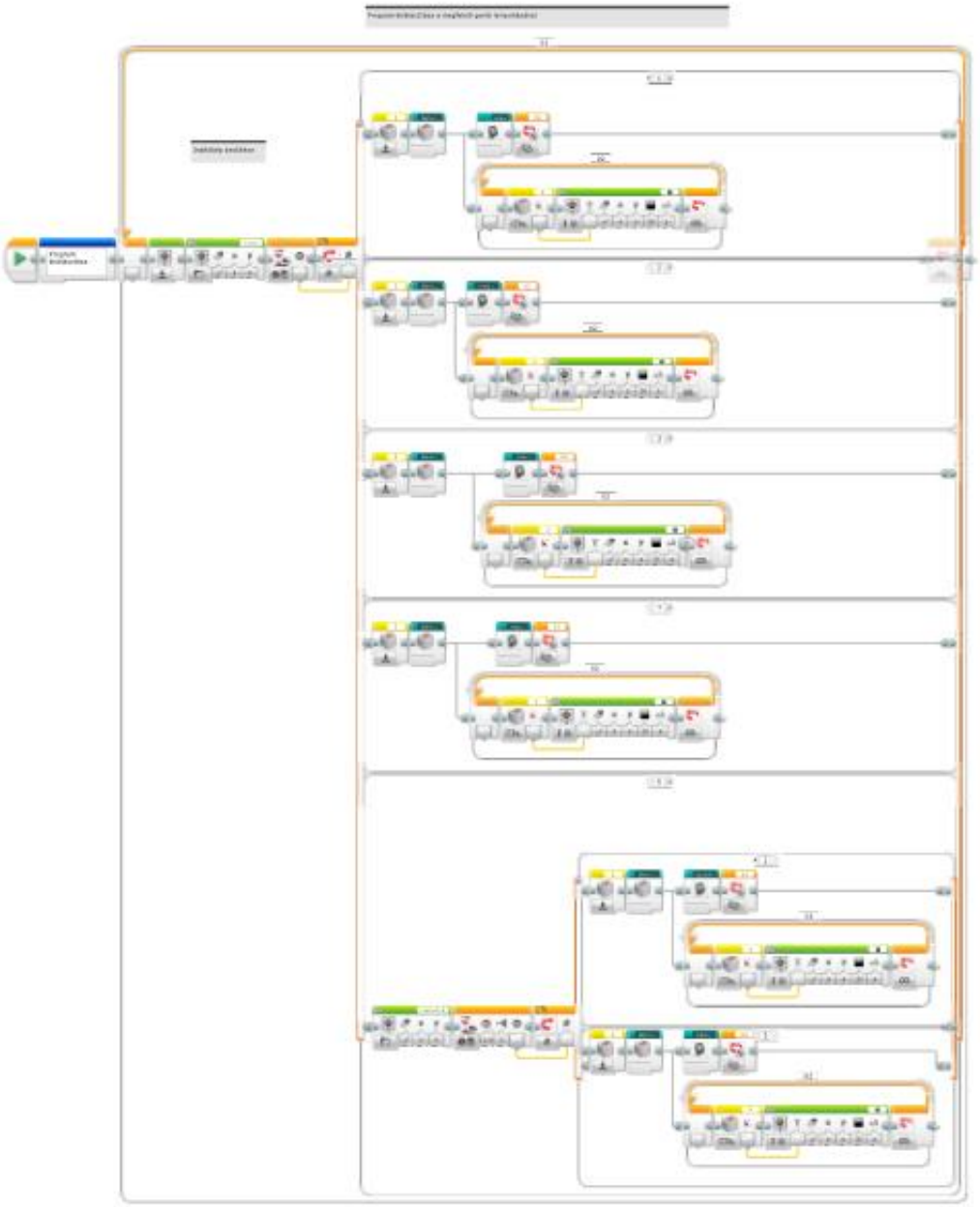


3. LAUNCH

We planned to solve the tasks in 6 rounds. For the run we created a master program, in which the rounds have been assigned to the buttons of the Brick.



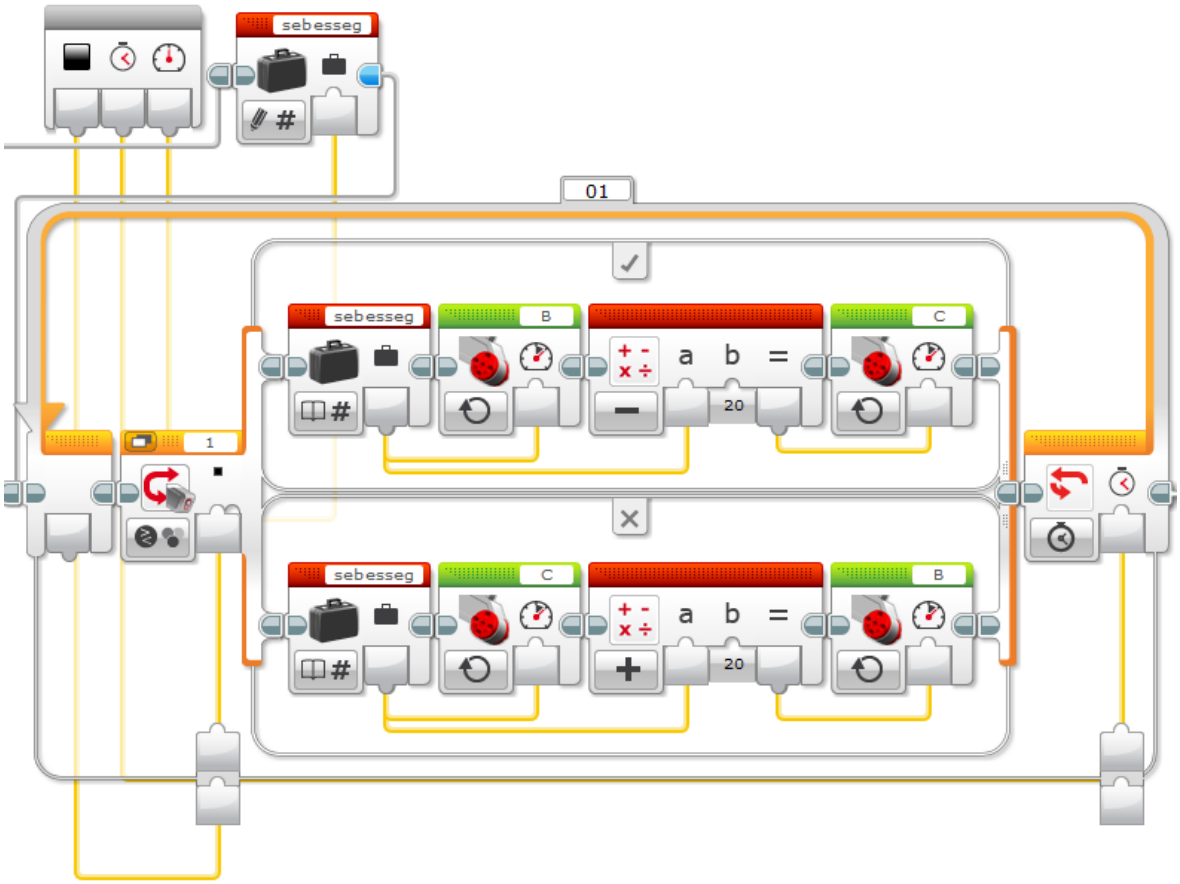
In the picture we can see the EV3 program, which contains a continuous gyroscopic measurement, which result we can always see in the screen. s



4. MYBLOCKS

4.1. LineFollow

4.1.1. Classic

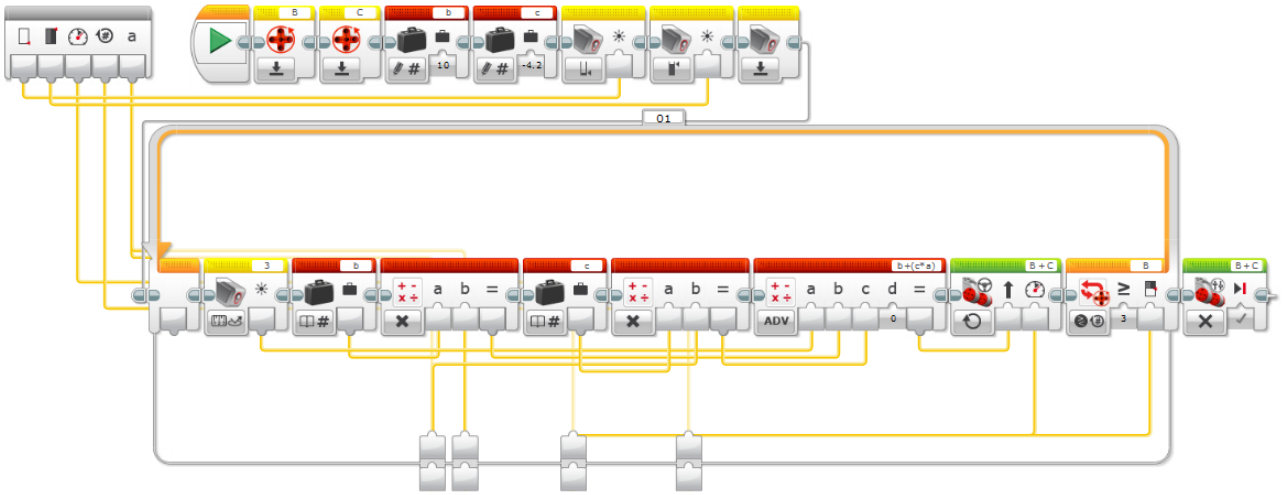


The block has 3 input parameters, that can be easily used by different tasks. The first parameter decide that, which color has to be followed by the robot, the second gives the time and the third is responisble for the velocity.

The program follows the gradient (colorchange), if the sensor perceives the given color, then turns parallel to the line, if it sees other color, then turns back to the line and during that goes ahead.

4.1.2. LineFollow with Calibration

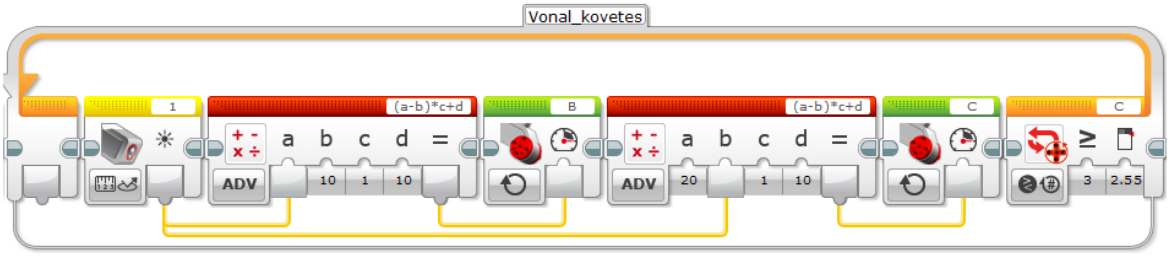
This MyBlock is measuring the reflected light. It calibrates the color sensor's minimum and



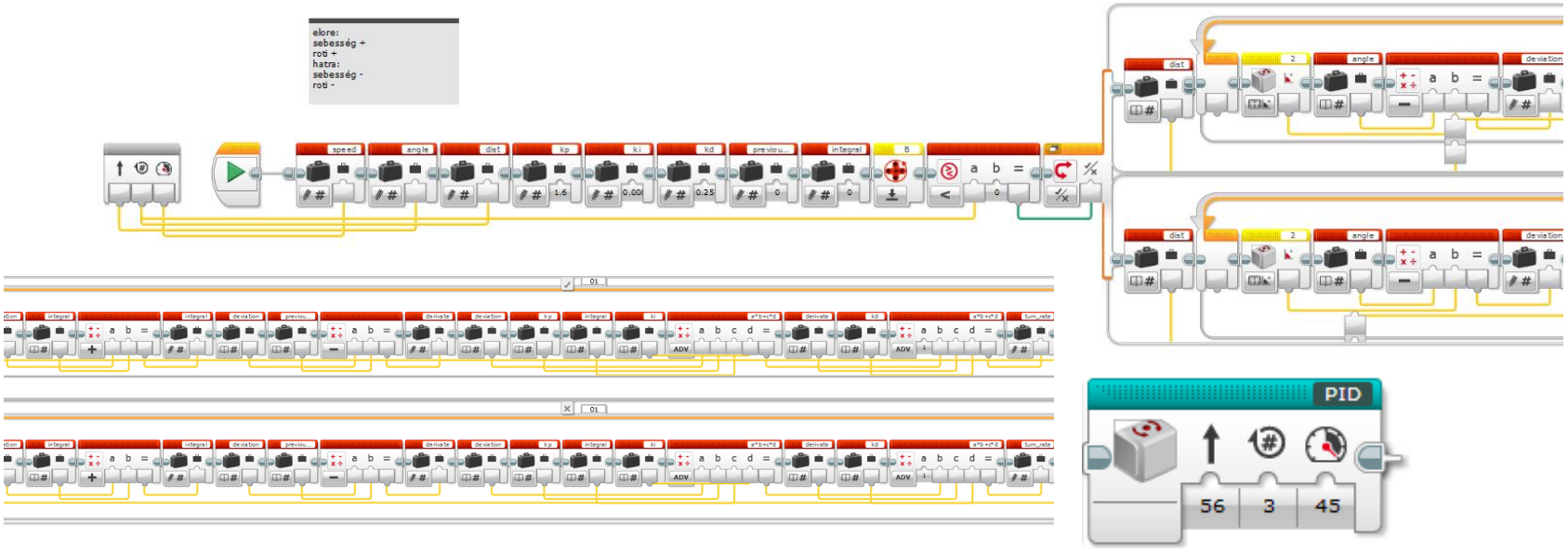
maximum perceptional limit form the given parameters at the begining part. In the loop we use the two edge parameters and the reflection of the light to tell the robot the right way.

4.1.3. Else

We created other line follower algorithms, where not the colors are observed, but the reflected light. Here we also follow the gradient.



4.2. PID

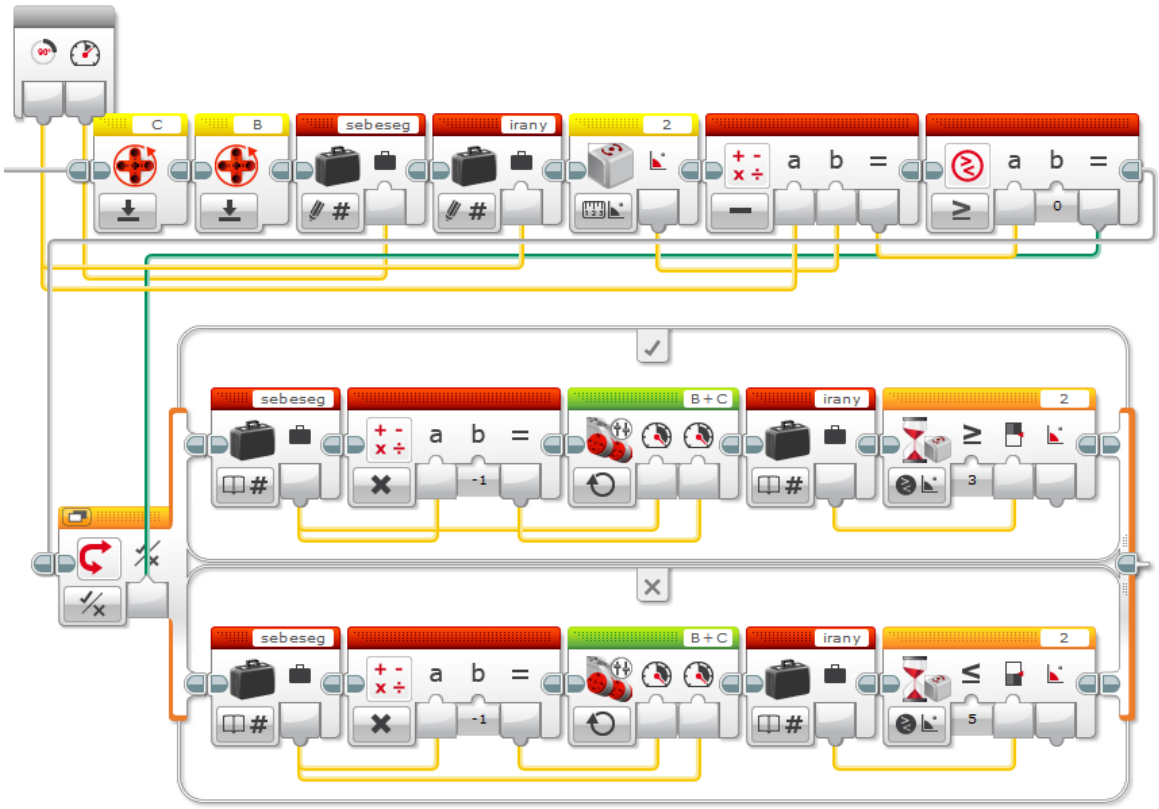


In the previous page you could find a subrutine of our program, which called PID. With the help of this corrector program, we can precisely move our robot forward.

To use the MyBlock we have to use 3 parameters. The first is the angle, the second is the required distance (in rotation), the third is the required acceleration. This block stores the previous error and with the help of it tries to correct the current error during the run.

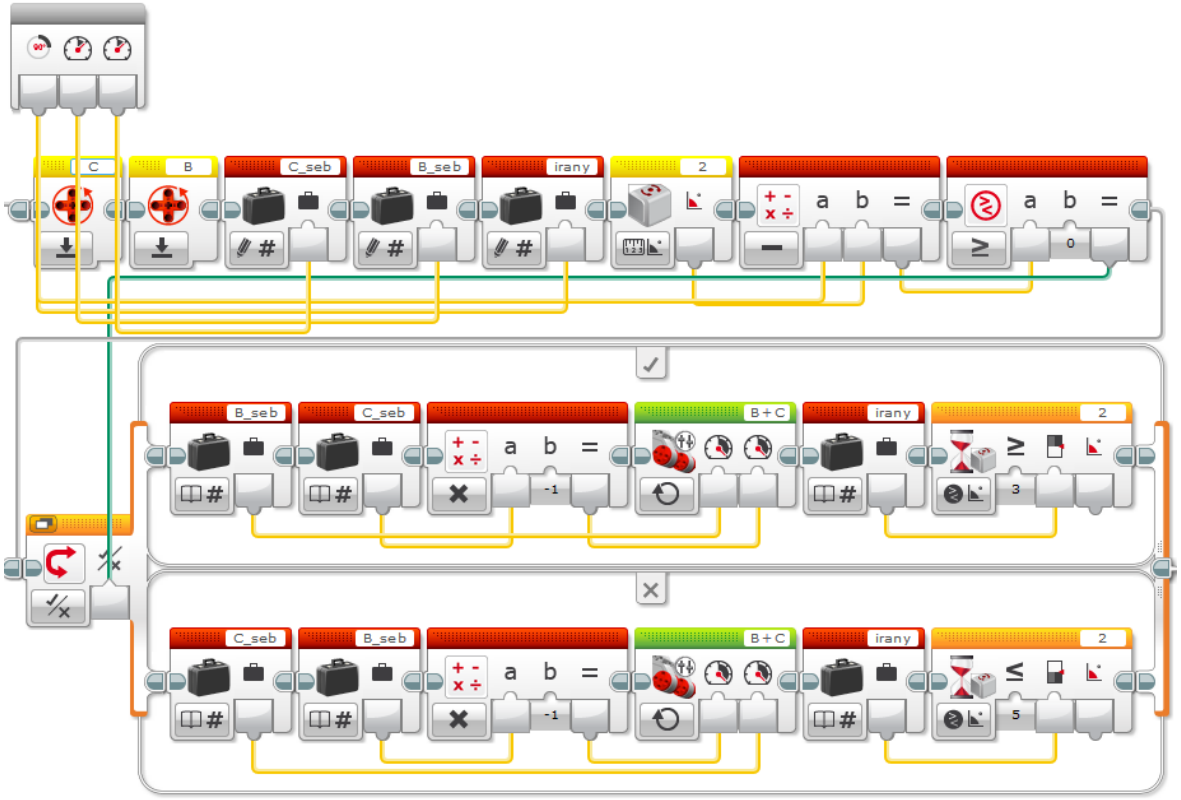
4.3. Rotate

4.3.1. With Two Wheels Equal Speed



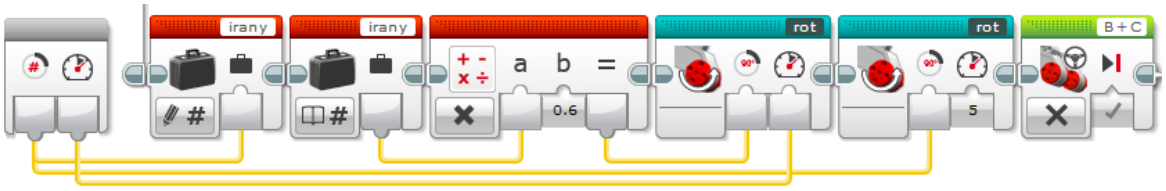
Two parameters needed to call the block: the angle of the turning and the velocity. Depending on the direction it moves one of the wheels forward and the other to backward, until it reaches the proper angle.

4.3.2. With Two Wheels Not Equal Speed



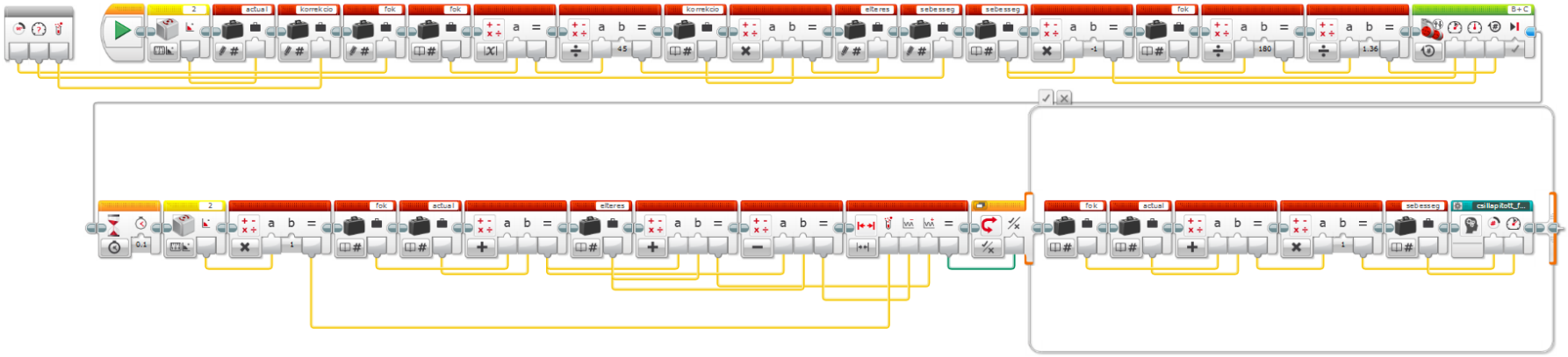
We filled out the previous block with another velocity parameter, that the wheels could move with different speed.

4.3.3. Attenuated turn



The block needs two parameters: angle and velocity. The block does the 60% of the turn with full speed and 40% with reduced speed. This block will turn with the help of the previous turning block.

4.3.4. Turn



The MyBlock that has been shown here is the one for our turnings. This routine has 3 parameters. The first stores the turning angle, the second is the velocity and the third saves the tolarance limit. The block built on the previously featured „attenuated turn”, however this program contains a selfcontrolling part. First we do a green block turn and after it has done, the gyroscope checks the success of the turning. If the gyroscope senses a difference, which is smaller than the tolarance limit, then corrects the turning with the attenuated turn. With the help of this program the robot could do relatively precise turns, when the gyroscope shows not appropriate value caused by any error.