LINE FOLLOWER ROBOT ------- Antonio Trujillo

General way how it works: First, it is constantly checking for the button being pressed. Once it is pressed, It will start moving, waiting to act depending on what the sensors encounter. This is done in order: firstly check for the marks, which tell the robot what way to choose during a split, then for the own split, after that, for any possible gaps, and finally the line to follow.

To summarize: at the beginning, the robot is waiting in the starting position, waiting for the button being pressed. When the button is pressed, it will follow the line until it encounter one mark indicating which way to follow when it detects a split. When the split is detected, depending on how the boolean variable was valued, it will follow the path indicated by the mark, or otherwise the other path. At the moment the split is detected, a variable will be activated, to let other functions know that the robot is going through a split. So that, if the outer sensors detect something, it will be ignored, as it is not a mark. Then, when the robot pass the end of the split, the variable will be back to 0, letting other functions know that the split did finish. If there is no gaps, the robot will be normally following the line, performing as precised if it encounters marks or splits, until it get back to the starting position, where it will stop when the outer sensors encounter the two marks. If any gap is reached, the robot will go on rotating from one side to the other until it reaches the line again. At that moment, the robot will turn to a specific side depending on which sensor has reached the line. After that, it will follow the line as normal.

Tricks:

For the splits, at the moment it detects one, it will ignore the undesired side for some time. So that, at a certain point, the robot can pass through the split following the correct path. By using the millis() function, which basically output the time passed since the start of the current program, it enters a loop that give the robot enough time to pass the split. Also, another variable is used to avoid the sensors detect a fake split in the first second of the program due to an issue in the starting position of the robot, so it will not be waiting for the end of it.

It ignores the wrong path using a variable that will be active if the start of the split is detected. That variable will be active until the robot reaches the end of the split, at that moment, it will be set to zero.

For the marks, if the variable that control the splits is not active, the variable that controls the marks will be set depending on the boolean variable that stablish if the path to follow is the one described by the marks or the inverse.

When the button is pressed being the robot at the starting position, the two marks are detected for the first time. After passing through them, a variable “ignore” is set to zero. So, when the robot makes one full lap, and get back to the starting position detecting the two marks, it will now stop there due the previous change in the variable.

The trick for the gaps is mainly making the robot move from side to side, and at some point, one of the external sensors will reach the line after the gap. In that moment, the robot will turn to the corresponding side so that it makes the central sensors reach the line, making the robot follow the line as normal. If one of the central sensors reaches the line before the external ones, it will follow the line by the own functioning of the code for previous tasks.

If all the sensors detect that there is white space (gap), the robot will perform certain instructions for specific amounts of time in loop. The instructions in the first loop are performed in the following order: straight(to make distance shorter), right and left, trying to reach the wider space possible. The robot is performing the instructions the time required by assigning one variable once with the millis() instruction and another one with the amount of time required for the instruction, so that we can control when that time has passed. While there is no line detected from any of the sensors, the instructions will be sequently performed in loop(after the first loop, the forward one will not be used). When it exits the loop, the robot will turn to one side, according to the sensor that closed the loop, until the line is reached. After this, all the variables are restarted, so that they are prepared for any possible future gap.