

Guided Mission with Reliability Techniques

for *FIRST* LEGO League Challenge



What techniques do you see in the video?

Aligning using walls

Aligning using mat marking

Proportional line follow for distance and until an intersection

Gyro-based turns

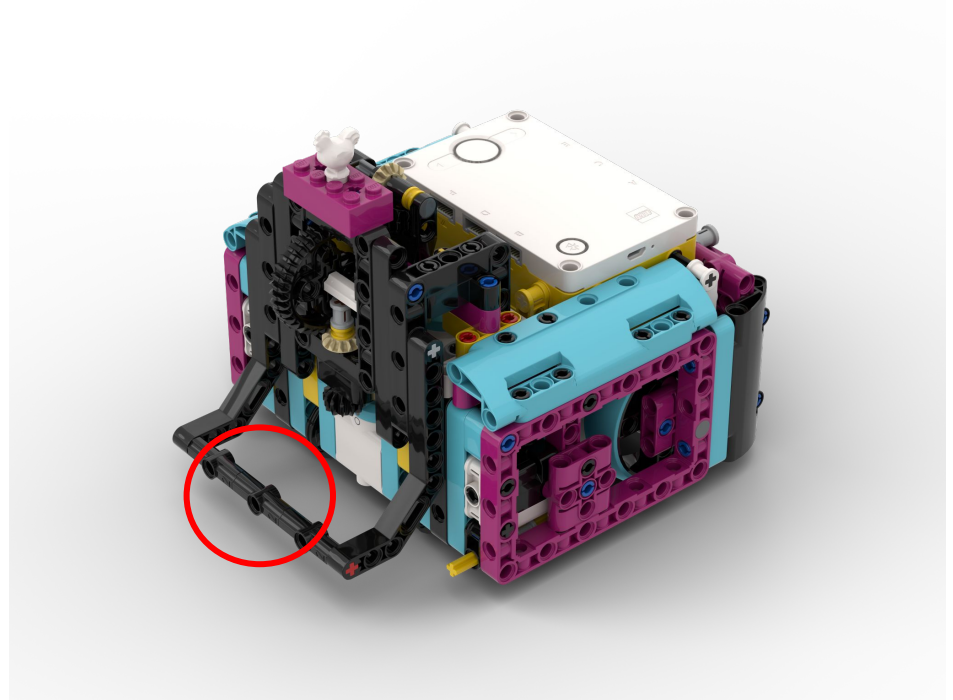
Squaring to a line to ensure robot is straight

Menu system to make choices

Building the robot in the video

These techniques can be used on any robot but if you want to build Coop Bot, building instructions are available on PrimeLessons.org and FLLTutorials.com

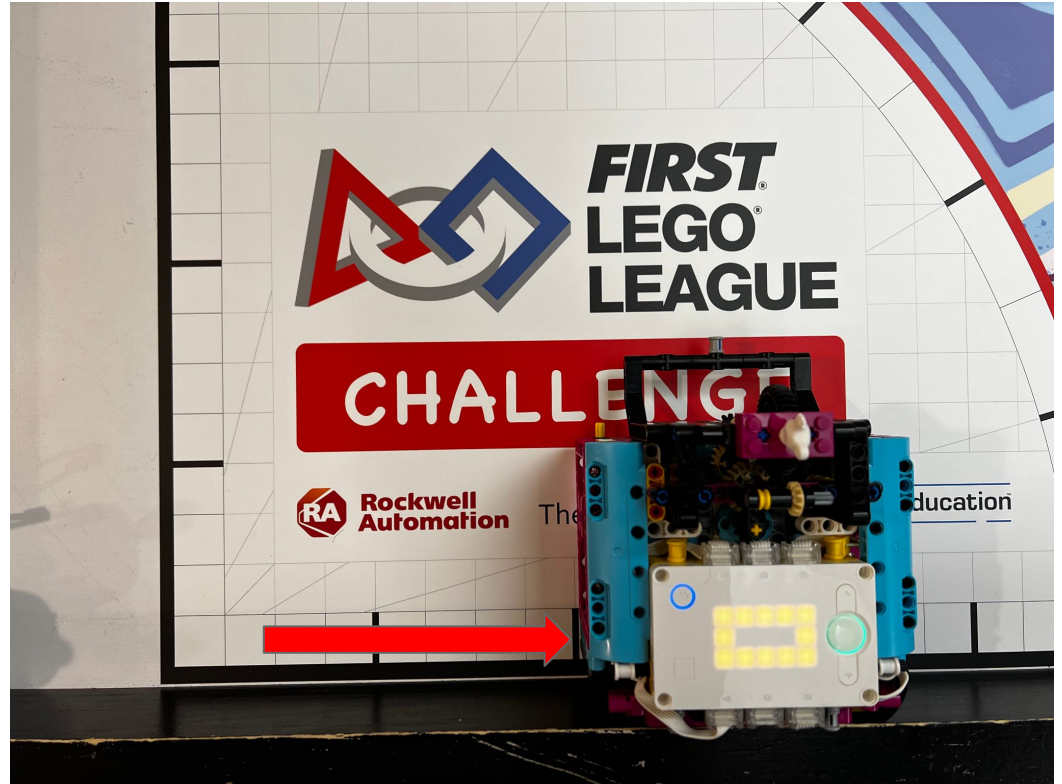
Note that a small modification was made to the universal attachment (indicated with a circle) for the shared mission



Alignment

Starting your robot against a wall or backing into the wall is important

Using the markings/images on the mat helps you with consistent starts



Proportional Line Follower

First, line follow for distance until the robot gets close to the intersection. This reduces the chance of the robot finding some color elsewhere on the mat that looks like the intersection

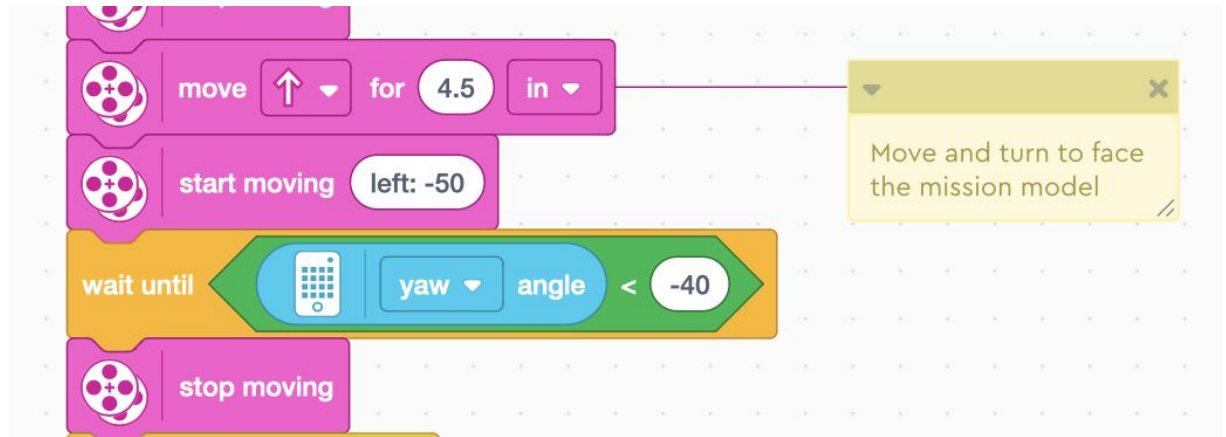
The second line follower goes until the intersection

It stops when the other color sensor detects the intersection

The image shows a Scratch script for a robot. The script starts with a blue block: 'F set relative position to 0'. This is followed by a green 'repeat until' loop with the condition 'F relative position > 400'. Inside this loop are three blocks: an orange 'set Error to' block with a light sensor icon and 'D reflected light - 50', another orange 'set Correction to' block with 'Error * 0.85', and a pink 'start moving' block with 'Correction'. Below this is a second green 'repeat until' loop with the condition 'A is color ?'. Inside this loop are three blocks: an orange 'set Error to' block with a light sensor icon and 'D reflected light - 50', another orange 'set Correction to' block with 'Error * 0.85', and a pink 'start moving' block with 'Correction'. The script ends with a pink 'stop moving' block. Two yellow callout boxes provide context: the first points to the first 'repeat until' loop and says 'Proportional line follow for distance. Follow the left side of the line since there is a white border only on that side'; the second points to the second 'repeat until' loop and says 'Proportional line follow until the other color sensor detects the intersection'.

Gyro-based turns

Turning using the gyro will be more accurate than turning using motor degrees



Squaring to a line

This is repeated multiple times to ensure that the robot is perfectly straight

The image displays a Scratch script for a robot's 'squaring to a line' routine. The script is contained within a 'repeat' block, which is partially visible at the bottom. The script consists of the following blocks:

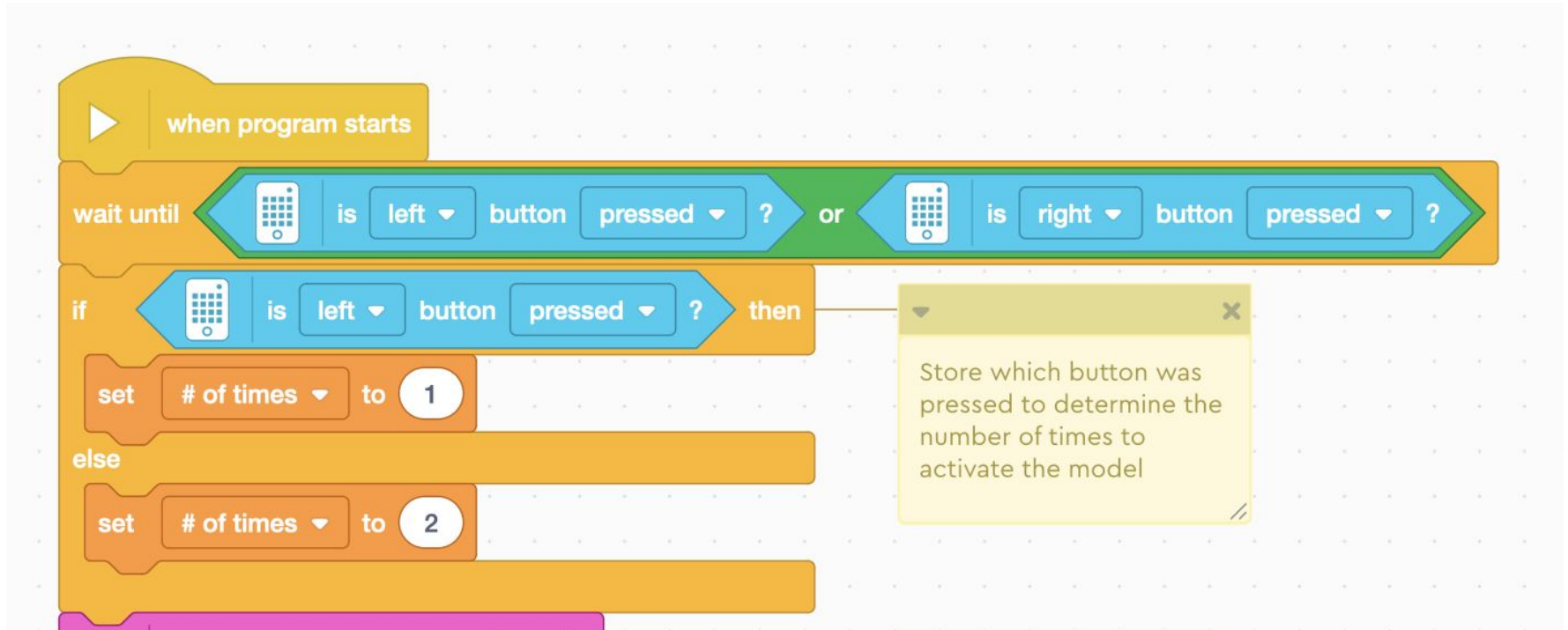
- stop moving** (pink)
- broadcast message1** (yellow)
- set speed to 25 %** (blue)
- start motor** (blue)
- wait until A is color black ?** (orange)
- broadcast message2** (yellow)
- start motor** (blue)
- wait until A is color white ?** (orange)
- broadcast message3** (yellow)
- start motor** (blue)
- wait until D is color black ?** (orange)
- stop motor** (blue)

A yellow tooltip box is positioned over the first 'wait until' block, containing the text: "Square on the line. Repeat three times to ensure robot is perpendicular to line".

To the right of the main script, there are three separate 'when I receive' blocks:

- when I receive message1**:
 - set speed to 25 %** (blue)
 - start motor** (blue)
 - wait until D is color black ?** (orange)
- when I receive message2**:
 - start motor** (blue)
 - wait until D is color white ?** (orange)
- when I receive message3**:
 - start motor** (blue)
 - wait until D is color black ?** (orange)
 - stop motor** (blue)

Menu lets you decide how many activations



The image shows a Scratch script on a grid background. It starts with a yellow 'when program starts' block. This is followed by a 'wait until' block with a blue arrow pointing right. The arrow contains two conditions: 'is left button pressed?' and 'or is right button pressed?'. Below this is an 'if' block with a blue arrow pointing right containing 'is left button pressed?'. The 'if' block has two paths: a 'then' path with a 'set # of times to 1' block, and an 'else' path with a 'set # of times to 2' block. A yellow tooltip box is connected to the 'if' block, containing the text: 'Store which button was pressed to determine the number of times to activate the model'.

Where can I learn to program like this?

All the lessons are available for free on PrimeLessons.org

Just like with this mission, you should combine techniques to complete other missions