


Guided Mission

for *FIRST* LEGO League Challenge


SUBMERGEDSM

Guided Mission

This season's Guided Mission is Send
Over the Submersible

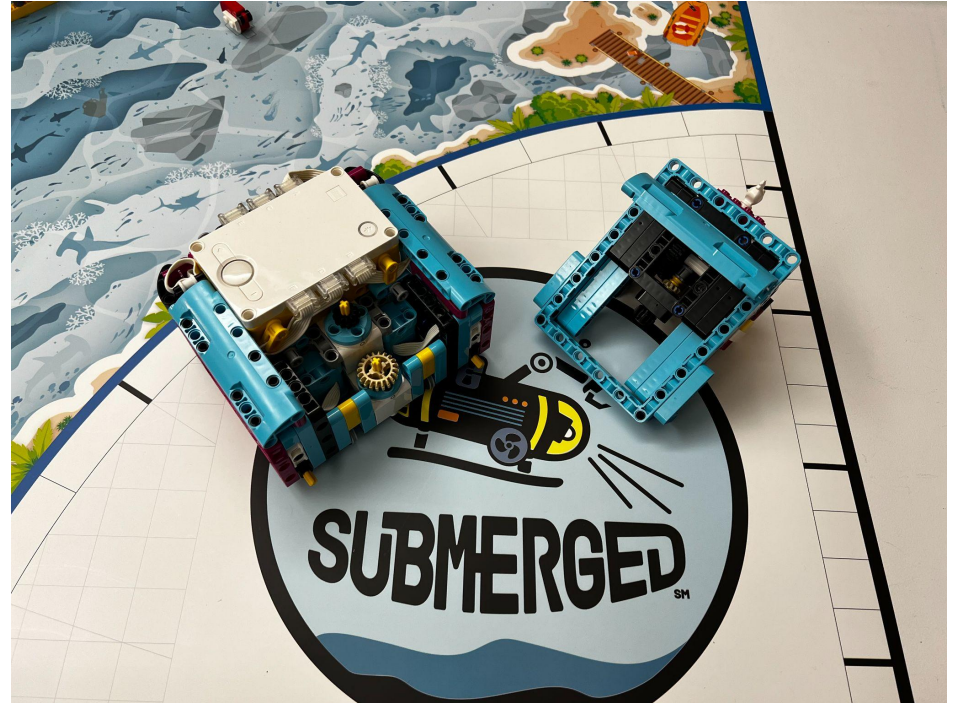


What Reliability Techniques are Used?

- Aligning using walls
- Aligning on mission models
- Aligning using mat marking
- Gyro-based turns
- Squaring to a line to ensure robot is straight

Robot Used in Video

- Notice that the mission requires teams to lift up a bar..
- Think about what you can built to do this.
- For this sample Guided solution, we use Coop Bot with a forklift.
- The techniques in this solution can be used on any robot but if you want to build Coop Bot, building instructions are available on PrimeLessons.org and FLLTutorials.com



Starting Position

- Position robot in the Blue Launch Area as shown
- Starting against a wall improves reliability
- Use the markings on the mat to be consistent



Other Techniques for Reliability

- While there are no lines to follow, there are lines to align on
- You can also align on mission models itself - be sure to create a flat bumper on your robot if you want to employ this technique



Pick a Strategy/Robot Path



Alignment points

Part 1

- This section of the code moves the robot from Launch Area up to the black and white lines
- Turning using the gyro will be more accurate than turning using motor degrees

The image shows a sequence of code blocks for a robot program. The blocks are as follows:

- when program starts** (yellow)
- set movement motors to B+F** (pink)
- set movement speed to 75 %** (pink)
- set 1 motor rotation to 17.5 cm moved** (pink)
- set yaw angle to 0** (light blue)
- C set speed to 50 %** (blue)
- C run for 50 degrees** (blue)
- move up for 9.75 in** (pink)
- set movement speed to 25 %** (pink)
- start moving left: -50** (pink)
- wait until yaw angle < -89** (green)
- stop moving** (pink)
- set movement speed to 75 %** (pink)
- move up for 13 in** (pink)

Callout boxes on the right side of the code provide additional context:

- Configure motors and reset yaw angle** (yellow)
- Move forward out of launch** (yellow)
- Turn using gyro to face the line** (yellow)
- Move forward towards line** (yellow)

Part 2

- Squaring on a line improves reliability and helps the robot straighten out as well as know where it is on the mat.
- This is repeated multiple times to ensure that the robot is perfectly straight
- Note, to make this code neater and more easily reusable, you could make it into a MyBlock.

The image displays three columns of Scratch code blocks, likely for a robot navigation task. The code is organized into three distinct sections, each triggered by a broadcast message.

Section 1 (Left Column): Triggered by 'broadcast message1'. It starts with motor B set to 25% speed and started. It then waits until sensor A is color (white), stops motor B, and waits 1 second. This sequence is repeated with sensor A set to black.

Section 2 (Middle Column): Triggered by 'broadcast message2'. It starts with motor F set to 25% speed and started. It then waits until sensor D is color (black), stops motor F, and waits 1 second.

Section 3 (Right Column): Triggered by 'broadcast message3'. It starts with motor F set to 25% speed and started. It then waits until sensor D is color (white), stops motor F, and waits 1 second.

Part 3

- Gyro-based turns used to turn towards the mission model.

The image shows a sequence of code blocks for a robot mission. The blocks are as follows:

- set movement speed to 25 %
- start moving right: 50
- wait until gyro yaw angle > 14
- stop moving
- set movement speed to 75 %
- move up for 18.5 in
- set movement speed to 25 %
- start moving left: -50
- wait until gyro yaw angle < -44
- stop moving

Three callout boxes provide descriptions for specific blocks:

- Turn using gyro to face towards the mission (points to the 'wait until' block with angle > 14)
- Move towards Guided Mission (points to the 'move' block)
- Turn using gyro to face towards the line (points to the 'wait until' block with angle < -44)

Part 4

- Square on the next set of black and white lines near the mission model.

The image displays three sets of Scratch code blocks, each representing a different scenario. The first set is triggered by a broadcast message 'message1' and involves motor B. The second set is triggered by receiving 'message2' and involves motor F. The third set is triggered by receiving 'message3' and involves motor F. Each scenario includes a wait block and a color-detection block.

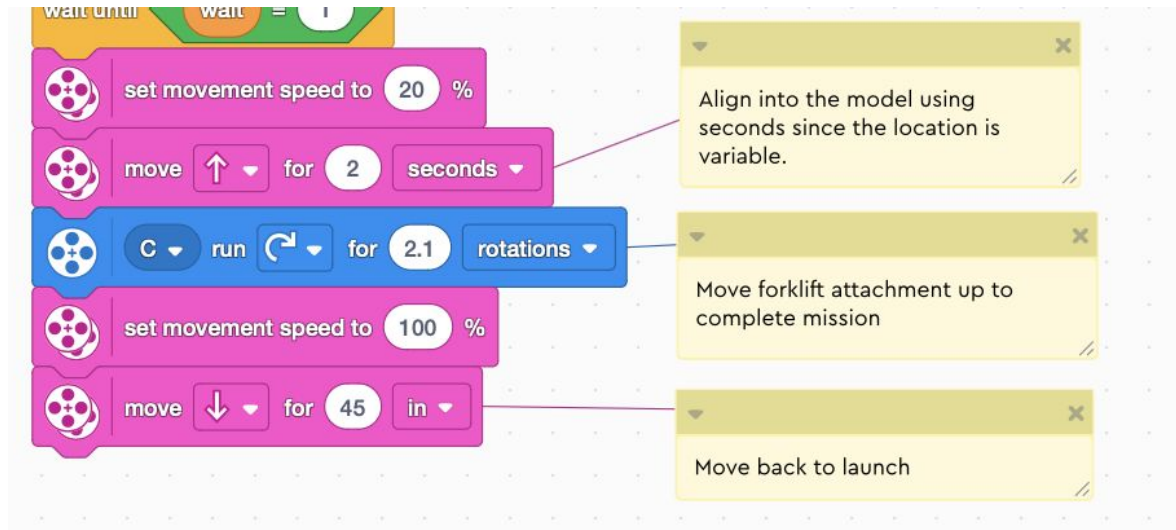
```
Scenario 1: Broadcast message1
- broadcast message1
- B set speed to 25 %
- B start motor
- wait until A is color (white)
- B stop motor
- wait until wait = 1
- broadcast message2
- B start motor
- wait until A is color (black)
- B stop motor
- wait until wait = 1
- broadcast message3
- B start motor
- wait until A is color (white)
- B stop motor
- wait until wait = 1

Scenario 2: When I receive message2
- when I receive message2
- set wait to 0
- F set speed to 25 %
- F start motor
- wait until D is color (black)
- F stop motor
- set wait to 1

Scenario 3: When I receive message3
- when I receive message3
- set wait to 0
- F start motor
- wait until D is color (white)
- F stop motor
- set wait to 1
```

Part 5

- Move into model (for Seconds) to align on it (important since model has variable placement on mat)
- Lift forklift to activate model
- Return back Home as quickly as possible using the shortest path



- Note that a Krill might be along the path. Make sure elements on the table do not go under the robot, and you can even collect it on the way back!

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 reliability techniques to complete other missions