Sense Autonomous Vehicle Algorithms

NeuLog

Sense Autonomous Vehicle Algorithms

2_5

For further information contact info@neulog.com

Contents

Chapter 1 – Autonomous Vehicle Algorithms

1.1 Autonomous vehicles

We are in the generation of autonomous vehicles, machine learning and artificial intelligence. This is the world of machines making decisions. The decisions are according to the software and programming behind. This is just the beginning.

We can understand this world and the occurring changes by trying to develop programs similar to autonomous car.

The SENSE is a tool for such challenge exercises.

This chapter introduces several of the challenge autonomous exercises. The idea is to let the user to think about algorithms and solutions to solve these challenges.

1.2 Program structure and algorithms

A control program usually has three parts:

- Initializations (setup)
- Procedures (sub routines) and functions
- Main program

The main program operates the procedures and the functions and runs in endless loop.

The procedures and the functions use variables that were defined in the initialization part. The values of the variables can be changed, of course during the program running.

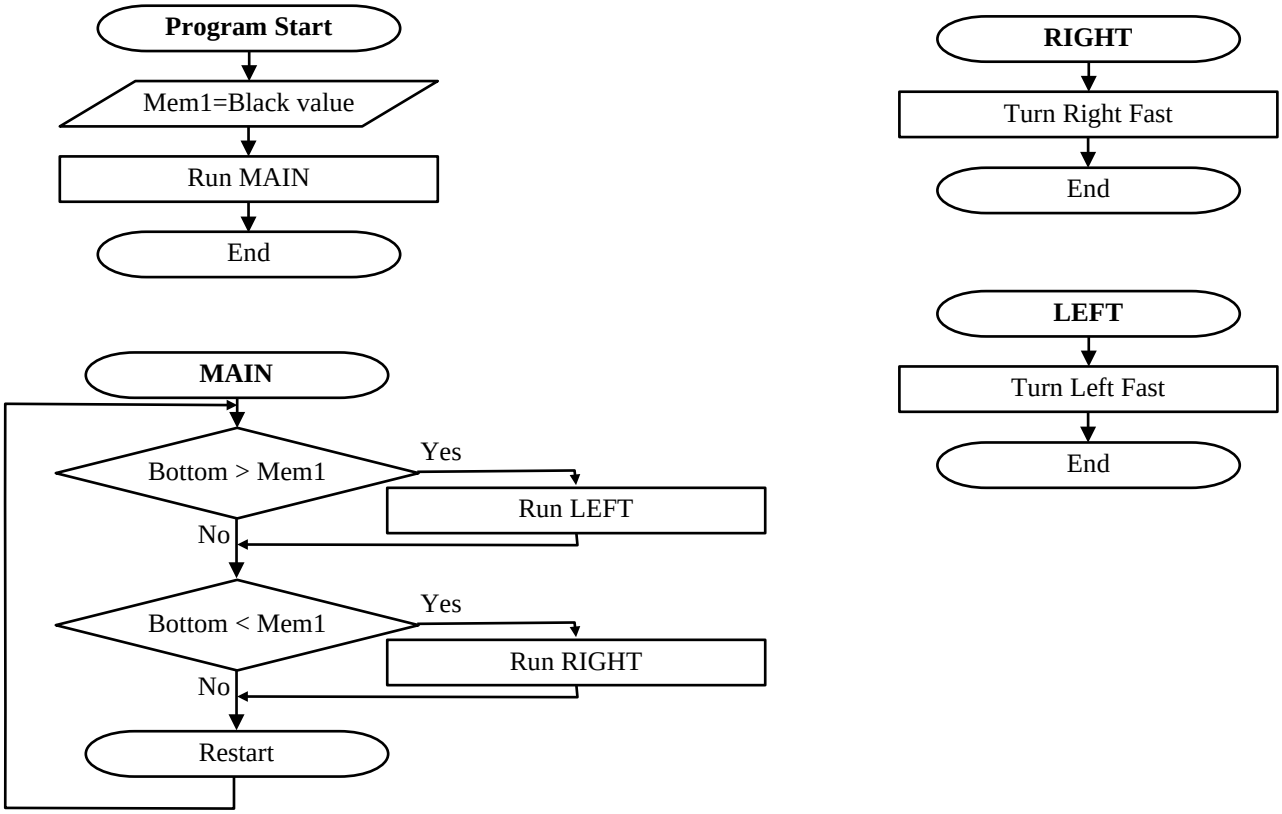
The procedures and the functions can also define new variables and use them.

We have to remember that every procedure and function returns to the instruction that follows the instruction, which calls it.

In RobocklySense we cannot work without procedures. This forces us to work properly.

An algorithm is the description and the methods of a program. We use flowcharts in order to describe the algorithm of a program. The flowchart is a map that gives a clear visual picture of the program.

The following is an example for moving along a black line program.



We must be aware from creating a long flow chart that makes the program complicated to understand.

1.3 Programming languages

The **RobocklySense** is the best program to start with. It is simple but very powerful. It forces the programmer to use variables (memories) and procedures, which makes the program structured and easily understood. This is also very important to the teacher.

With **RobocklySense**, one screen program can be a solution for a complex challenge mission.

For other programming languages such as: **Blockly**, **Python**, **C language** or **Arduino,** we can plug coding units into the SENSE.

Each coding unit comes with user manual describing how to work with it.

This chapter is built as challenge exercises to solve with general guiding and flowcharts.

There are multiple solutions. Try to find the most efficient way.

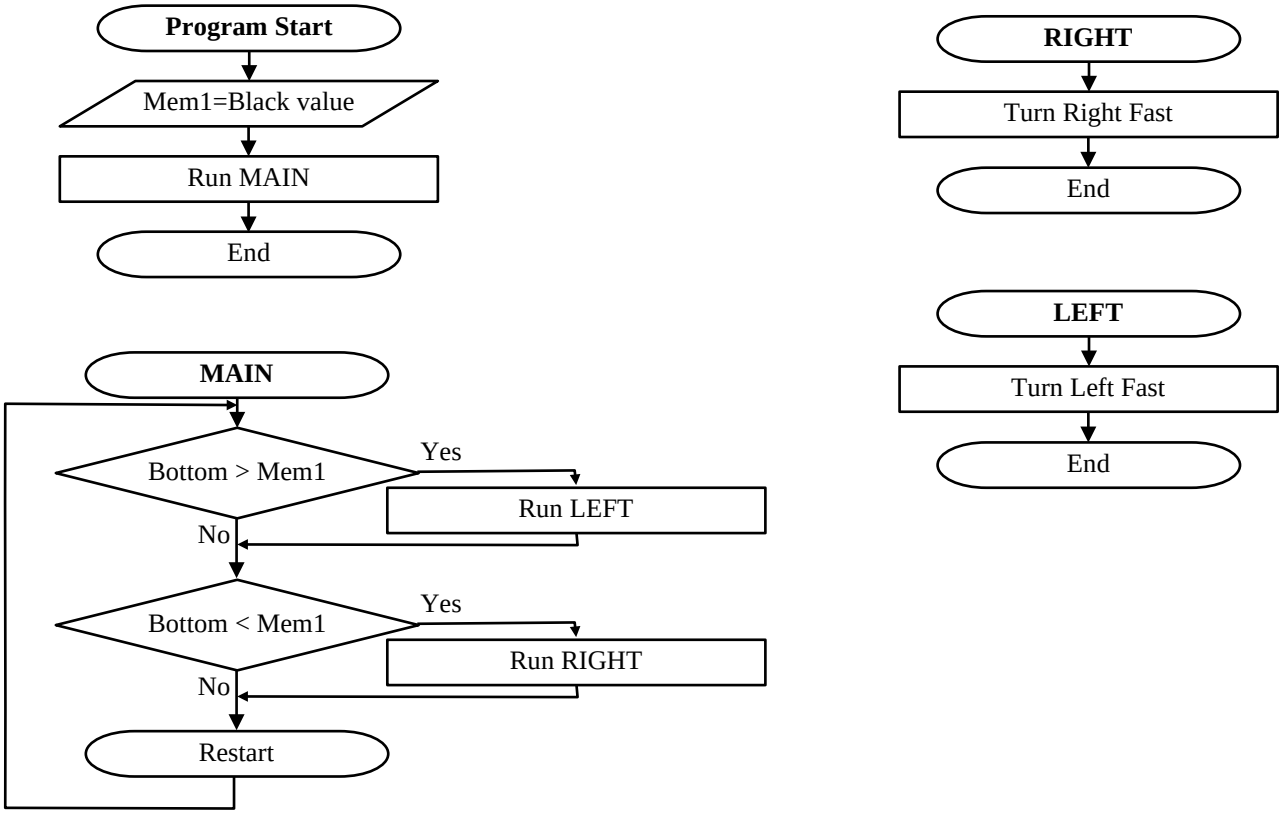
Do not be afraid to fail and to try again and again.

Good Luck!!

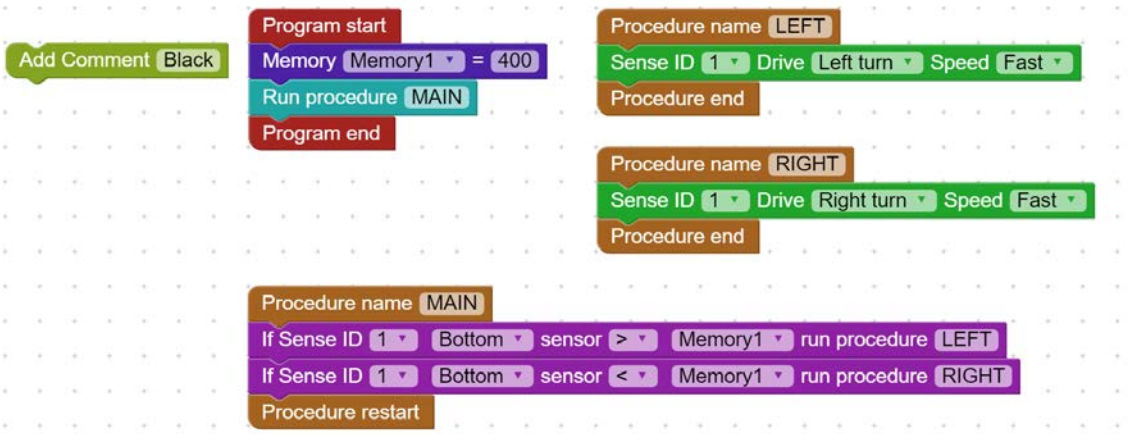
Challenge 1.1 – Along black lines

1.1.1 Left and right along a black line

The following is the flowchart of a simple movement along a black line program.



The following is RobocklySense program for the above flowchart.



The robot moves by swinging on the edge of the black line.

We use the RobocklySense **Direct** mode to find the **Black** value for the memory1 definition.

Place the robot on the black line, read the bottom sensor value and set it in the program. This value may be different from one robot to another.

Download, run and check to robot movement.

1.1.2 Smooth movement along a black line

In order to get a smoother movement we can replace one of the turn instructions with a deviate instruction. This depends on the robot movement direction.

When the robot moves counter clockwise, we shall replace the **Right turn** command with **Right deviate**.

When the robot moves clockwise, we shall replace the **Left turn** command with **Left deviate**.

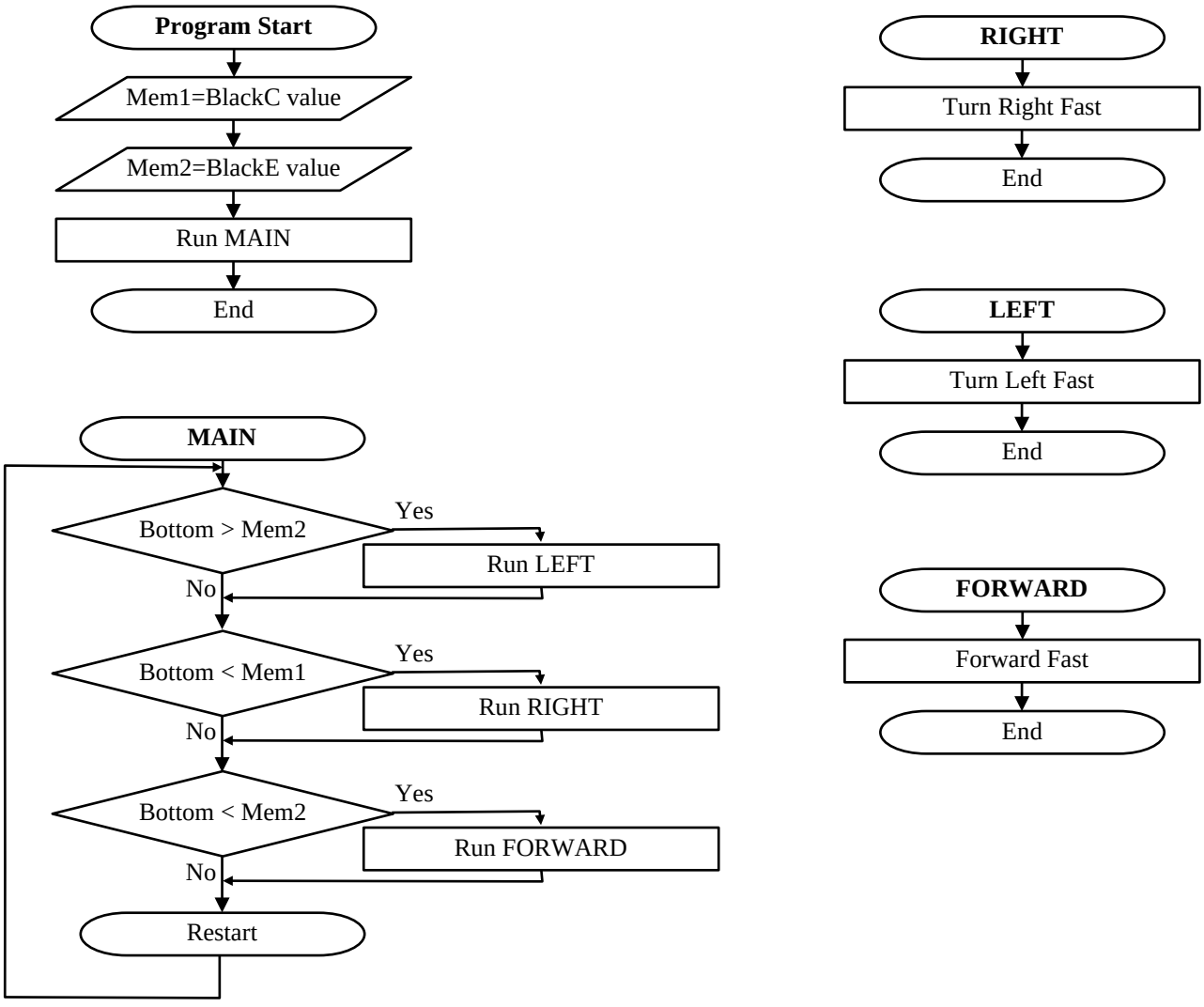
Change the program accordingly, download, run and check to robot movement.

1.1.3 Adding Forward movement

Check at **Direct** mode, the bottom sensor value when it is above the center of the black line and when it is closer to the edge of the line.

We shall call the read value at the center of the black line **BlackC** and the black value close to the edge **BlackE**.

The following program drives the robot forward when it is on the edge part of the black line.



Analyze the flowchart.

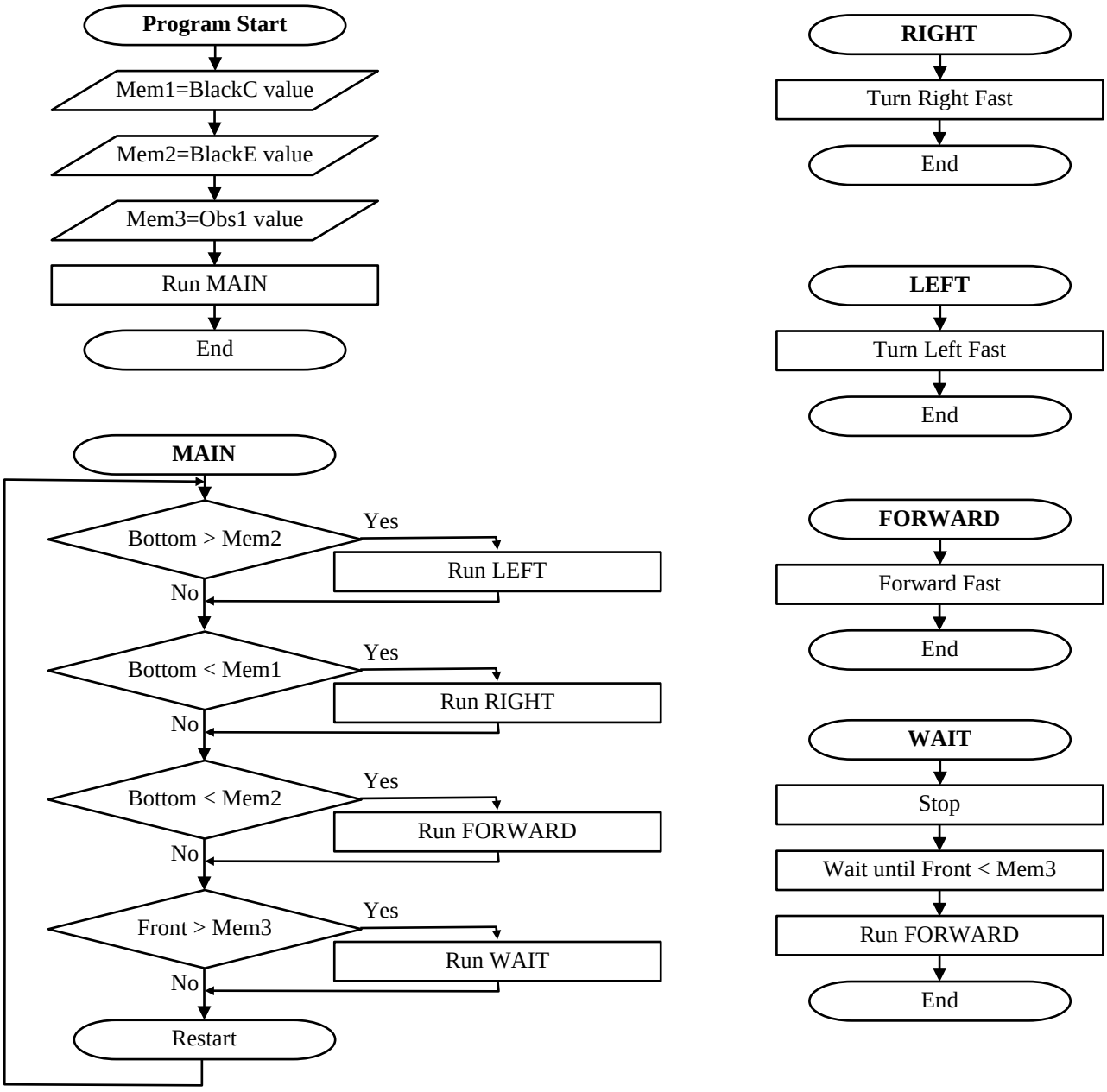
Change the program accordingly, download, run and check to robot movement.

1.1.4 Along a black line with a stop in front of an obstacle

Improve the previous program to stop in front of an obstacle until the obstacle is removed.

Put your hand in front of the SENSE and check at **Direct** mode, the front sensor value. We shall call the read value Obs1.

The following program drives the robot forward when it is on the edge part of the black line.

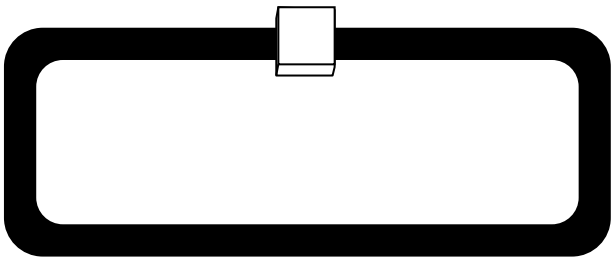


Analyze the flowchart. Change the program accordingly, download, run and check to robot movement.

Challenge 1.2 – AGV – Automatic Guided Vehicle

An AGV is a vehicle or a cart that moves along guidelines. It is very popular in manufacturing places for transporting row materials or sub-assembly systems from one station to another.

Create the following line.



Put a small box on it as in the picture.

Write a program that moves the SENSE along the line and stops in front of the box for 5 seconds, turns around, moves on the other direction and vice versa.

The SENSE goes on the outer edge.

For this task we have two movements – clockwise and counter clockwise.

The main program should know what the current movement is. To determine that, we use what we call a flag. The main program operates the required procedure according to the value of a certain variable.

The value of this variable is changed when changing direction is needed.

Analyze the following flowchart. Memory 4 is the flag variable.

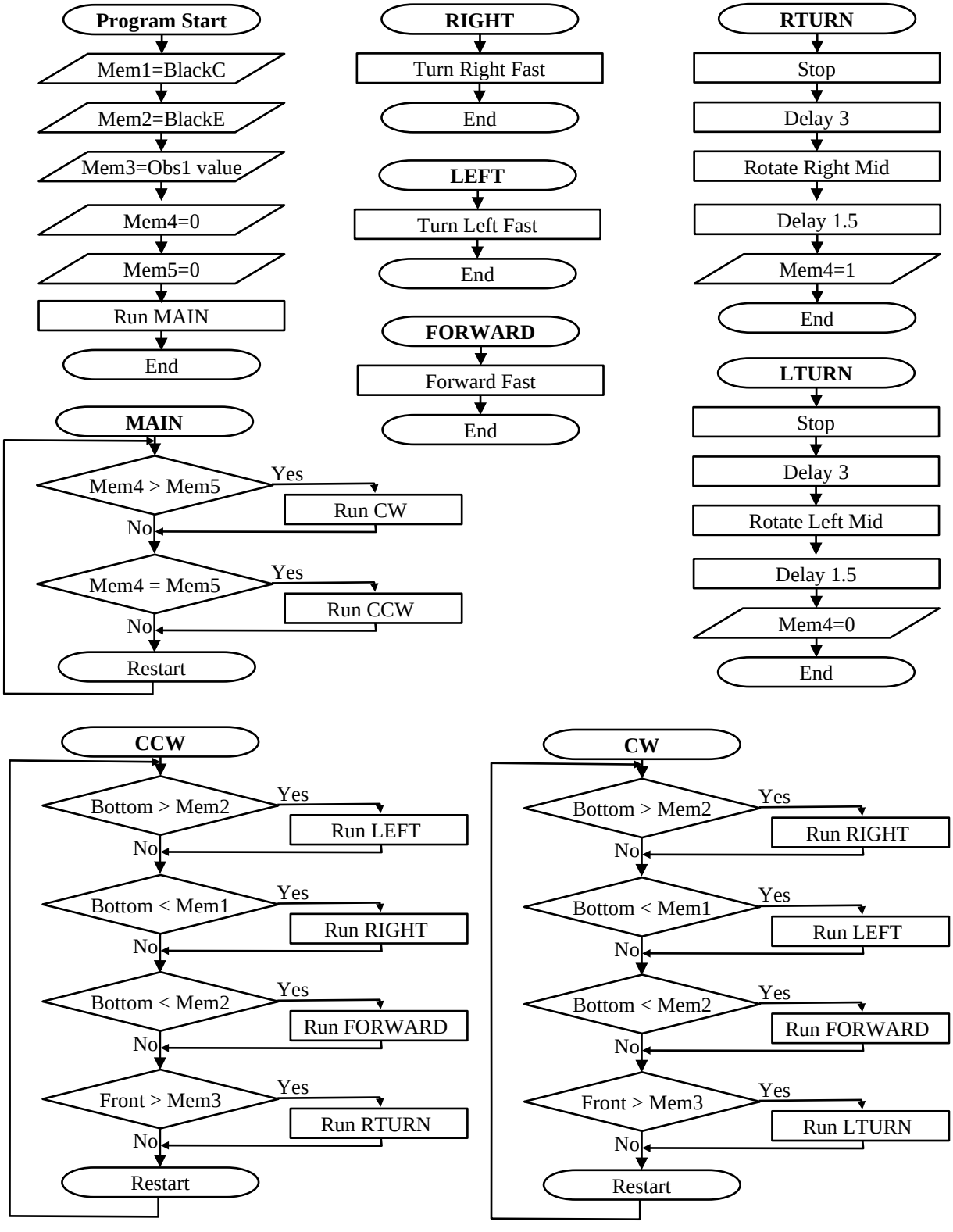
Build the program accordingly, download, run and check to robot movement.

Every robot behaves a little different.

Adapt the program to your robot sensors and behavior.

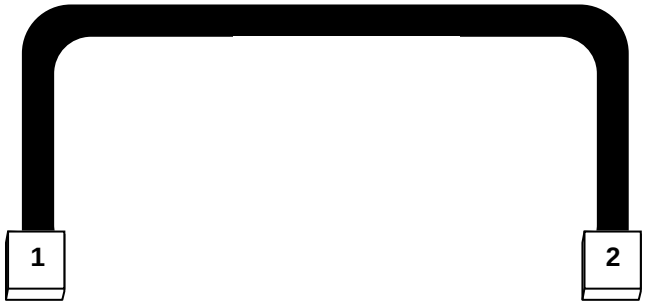
Take care to stop in front of the box in a distance that enables the robot to rotate.

AGV Program flowchart



Challenge 1.3 – AGV between stations

Create the following line with the boxes.



Write a program that moves the SENSE from one station to another along the lines in this order: 1-2-1-2-…

The Sense stops at each station and moves to the next station when you put your hand close to the right back sensor.

Hint:

The previous AGV program should answer the movement of the robot.

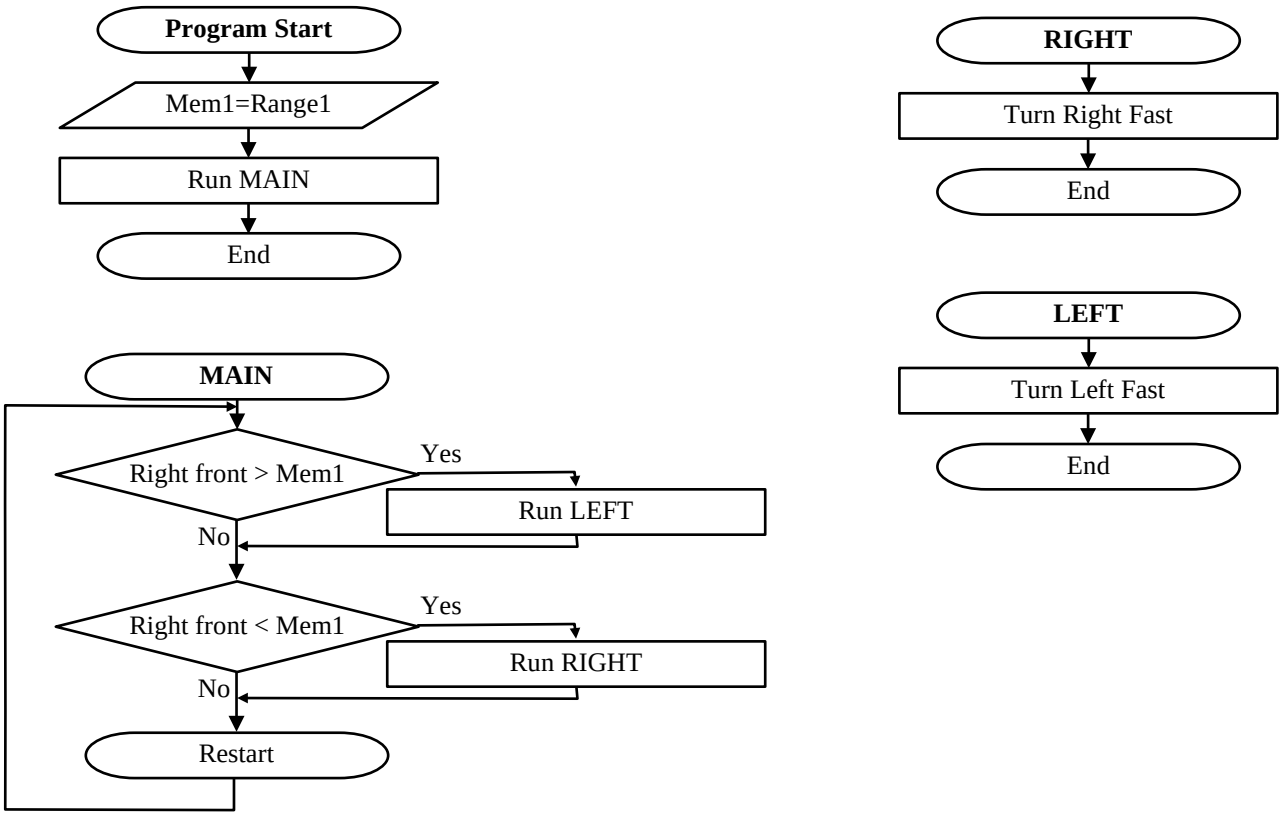
Challenge 1.4 – Along a building block

The following exercises deal with movement methods along walls and around a building block.

Use at least 40 X 40 cm box as a simulation of the building block.

1.4.1 Left and right along walls

The following is the flowchart of a simple movement along walls program.



The robot moves by swinging along a wall on its right side.

We use the RobocklySense **Direct** mode to find the **Range1** value for the memory1 definition.

Place the robot near the box on its right side, read the right front sensor value and set it in the program. This value may be different from one robot to another.

Download, run and check the robot's movement.

1.4.2 Smooth movement along a black line

In order to get a smoother movement we can replace one of the turn instructions with a deviate instruction. This depends on the robot movement direction.

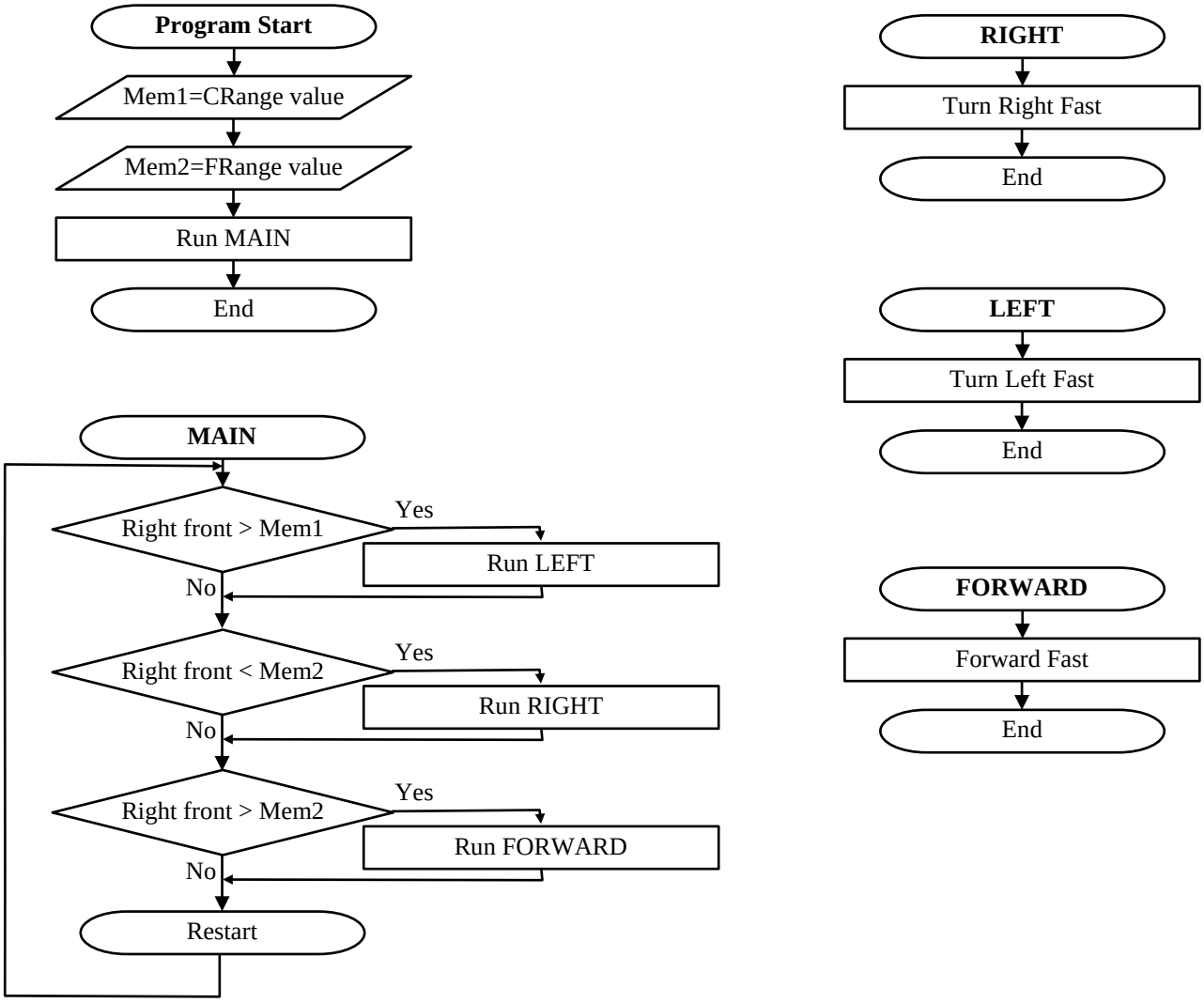
With the above program, the robot moves clockwise, we shall replace the **Left turn** command with **Left deviate**.

Change the program accordingly, download, run and check to robot movement.

1.4.3 Adding Forward movement

We can use two range values – **Crange** (close range) and **FRange** (far range).

The following program drives the robot forward when it is between CRange and FRange.



Analyze the flowchart.

We have to remember that the right front value increase when the robot come closer to the wall.

Determine the **CRange** value as the **Range1** value of the previous program.

Determine the **Frange** value as **CRange** – 10.

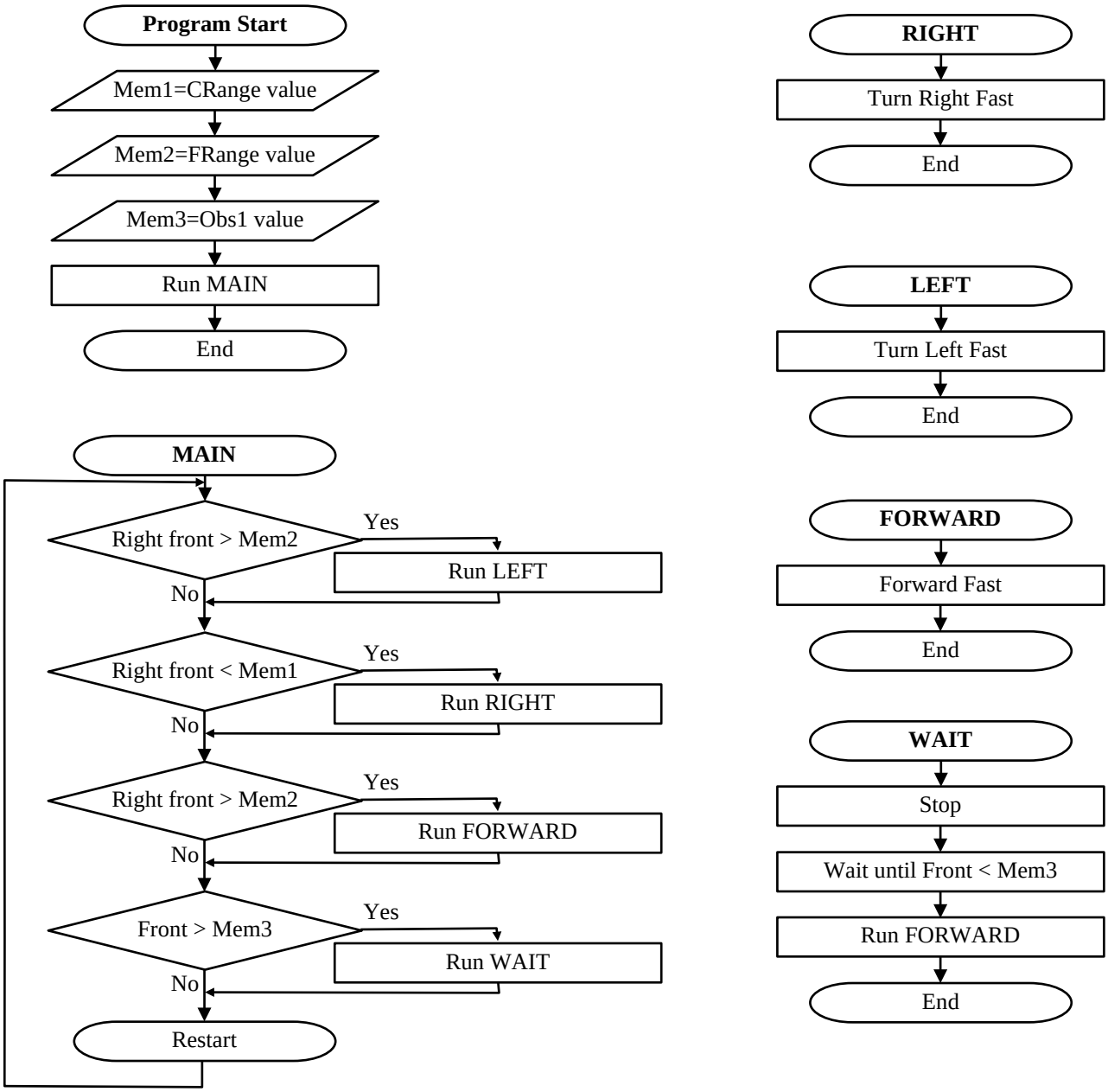
Change the program accordingly, download, run and check to robot movement.

1.4.4 Along a wall with a stop in front of an obstacle

Improve the previous program to stop in front of an obstacle until the obstacle is removed.

Put a small box in front of the SENSE and check at **Direct** mode, the front sensor value. We shall call the read value Obs1.

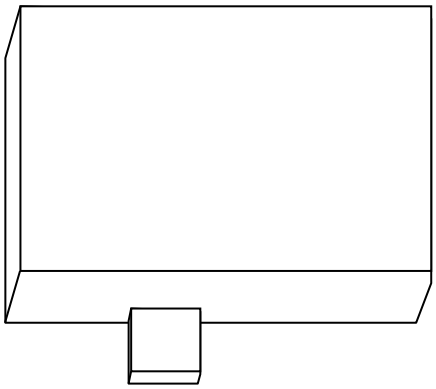
The following program drives the robot forward when it is on the edge part of the black line.



Analyze the flowchart. Change the program accordingly, download, run and check the robot's movement.

Challenge 1.5 – Along a building block and bypass cars

Put an obstacle, as described in the following picture.



Write a program, as in challenge 1.4.4, with SENSE bypassing the car. The SENSE should return to the right only after passing the car.

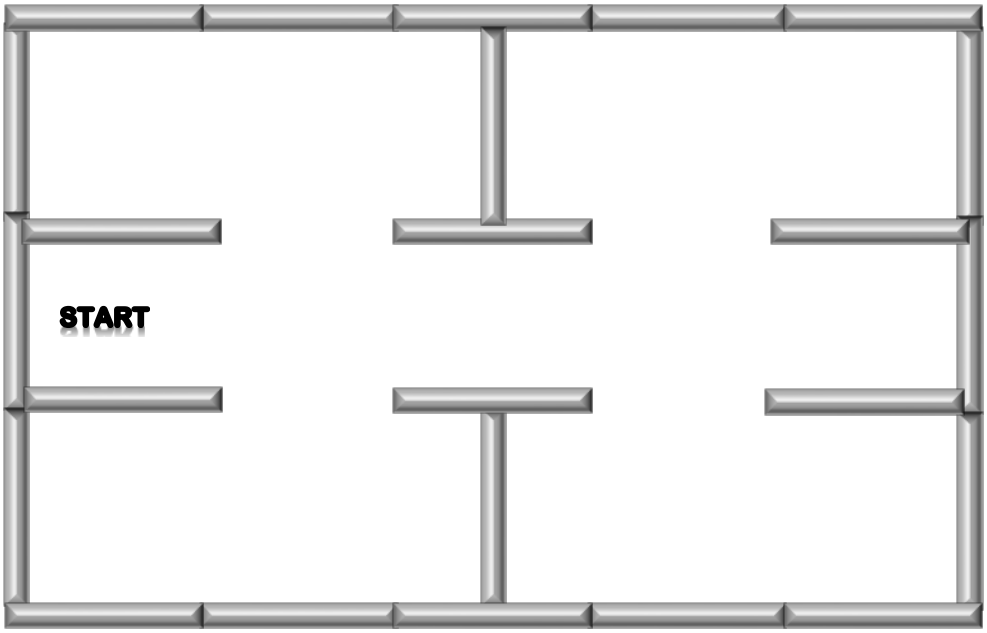
Hint:

Replace the **WAIT** procedure with **TURN** procedure.

The **TURN** procedure rotate to the left until the front sensor is below the **OBS1** value.

Challenge 1.6 – Autonomous museum guard

Build a model of a museum with rooms and corridors as follows:



Create a program that moves the robot along the walls through the museum rooms. The starting point is at the START position.

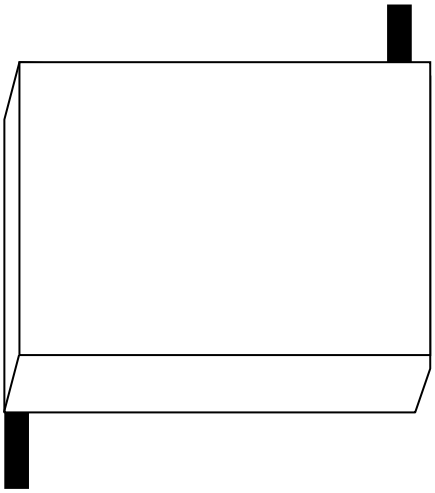
Hint:

The program of challenge 1.5 can serve as a solution for this task.

You have to adapt the memory values to the model.

Challenge 1.7 – Along a building block with stop sign

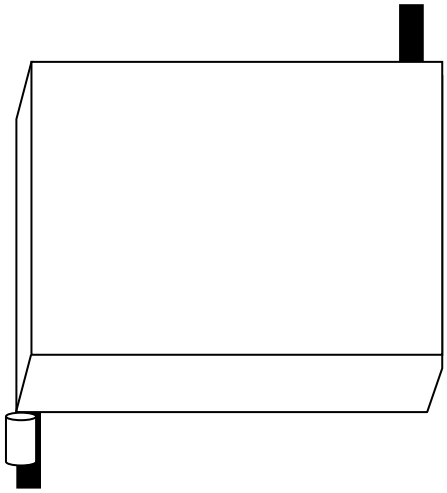
Put black lines at the corners, as described in the following picture.



Write a program, as in challenge 1.4, with SENSE stop at the black line for 3 seconds.

Challenge 1.8 – Along a building block with stop for pedestrian

Put a rod (simulates a pedestrian crossing) at one of the corners, as described in the following picture.



Write a program, as in challenge 1.4, with SENSE stop at the black line for 3 seconds.

It does not move on if an obstacle is in front of it.

Challenge 1.9 – Building block guard

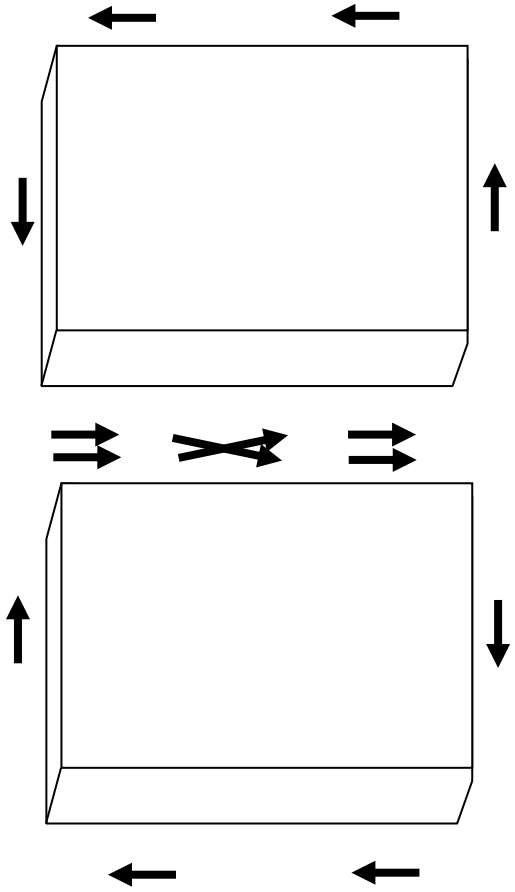
Write a program, as in challenge 1.4, with SENSE stop for 10 seconds after each round.

The program has to count the corner turns.

Hint:

- The SENSE has two range sensors on each side.
- The movement along the wall is according to the front side range sensor.
- When the robot is in a corner, the back side sensor moves away from the wall.
- The program should check the value of the back side sensor.
- When the value is low (the sensor is far from the wall), the robot should call a turn procedure with increasing the corner number.
- After counting four corners, the robot should stop for 10 seconds and then starts again.

Challenge 1.10 – Two buildings guard

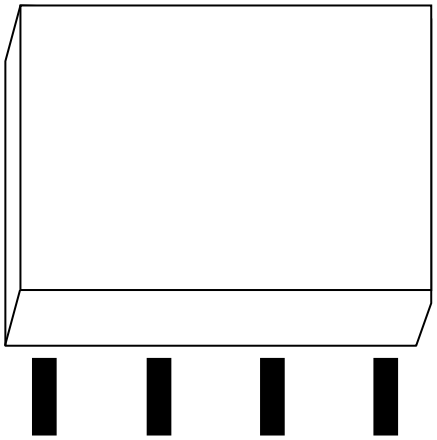


Write a program for the SENSE to go around the two buildings as in the above picture. The robot starts at the road between the two blocks.

The program has to count the corner turns and to change from moving counterclockwise around one building to clockwise around the other building.

Challenge 1.11 – Taxi driver

Put black lines along the building, as described in the following picture.

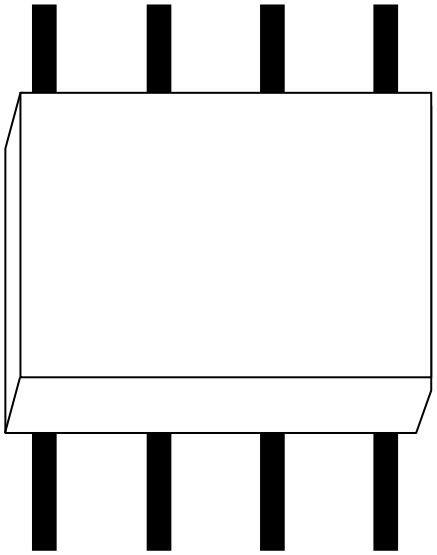


Write a program that moves the SENSE along the building and stops it on the third black line.

Start with the SENSE on the other side of the building.

Challenge 1.12 – Taxi driver with passenger

Put black lines along the building, as described in the following picture.



Write a program that moves the SENSE along the building and stops on the third black line for 5 seconds.

After that, the robot continues to the other side and stops on the second black line.

Challenge 1.13 – Home vacuum cleaner robot

Build a model of a room as follows:

Create a program that moves the robot along the walls in different distances from the walls.

At the first round, the robot will move closer to the walls and at the second round, the robot will move 8cm from the walls.