Arieh Nachum

SENSE Autonomous Robot

NeuLog

Arieh Nachum

SENSE Autonomous Robot

2_3

■ **www.ses.co.il** ■

Contents

Chapter 1 – Control and Robots

1.1 Robots

The world today is a world of embedded computer systems. We find them in media systems, watches, phones, remote control, cars and many more. Few years ago, we could not understand terms such as 'wearable computing' or 'internet of things'.

Everyday a new surprising product or application appears and months later, we cannot realize how we lived without it.

The robotic systems part of the embedded computer systems are systems that perform independent activities like search, manipulation, identification, activation, protection and so on.

Many systems combine a certain kind of artificial intelligence in operating and communication between machines.

Each robotic system includes a controller that allows it to operate in accordance with different operating programs. The robot developer writes these programs on a computer and forwards them to the controller.

The robotic system includes the controller, building components, wheels, gears, motors, sensors and more.

Building a robotic system creates a challenge to acquire knowledge in various technology areas (electronics, computers, mechanics, electricity, etc.).

There are many types of robots such as arm robots, mobile robots, walking robots and more.

The SENSE robots are a series of robots and "brain" units for study, programming and making robots with wide variety of robot applications.

The SENSE robots are autonomous robots and enable us to program many robot applications and functions. Each robot offers additional programming challenges such as: movement on a line, movement along walls, searching, standing, walking, etc.

1.2 Control systems

A robot is a computerized control system.

A "Control system" may be defined as a group of components, which can be operated together to control various variables, which govern the behavior of the system.

Air-conditioning system controls the room temperature.

A greenhouse control system controls temperature, humidity, light and irrigation.

A speed control system maintains a steady motor speed regardless of the changing load on the motor.

A light control system can maintain a steady level of light, regardless of the amount of available sunlight. The control system turns lamps ON or OFF according to the required light level.

Three basic units are in every computerized control system:

1. **Input unit** – the unit that reads the system sensors like temperature, light, distance, touch switch, etc. and feeds information into the control unit.

2. **Control unit** – the "BRAIN" of the control system, which contains the system program in its memory and performs the program instructions and processes the received data.

3. **Output unit** – the unit that operates the system actuators such as motors, lamps, pump, and fan operated as the results of the inputs and the program "decisions".

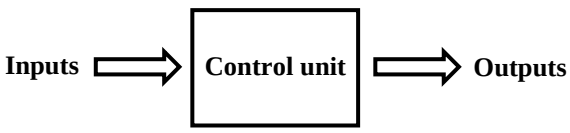


Figure 1-1

The control unit is connected to a computer for programming and downloading a program from the computer to the control unit flash memory.

Disconnecting the control unit from the computer and connecting power source or a battery to it, creates an independent system.

1.3 SENSE autonomous

SENSE autonomous is a mobile robot for applications such as:

- Movement along black line or white line
- Movement along walls or in a labyrinth
- **Autonomic car** – movement in a labyrinth when other robots are also moving there
- Following a moving body holding IR transmitter using tracking module
- Tracking and hitting a moving body holding IR transmitter using tracking module
- **Environment monitoring** and measurement robot with NeuLog sensors
- **Robot games** such as: football, catch me if you can, fighting robots

The SENSE includes:

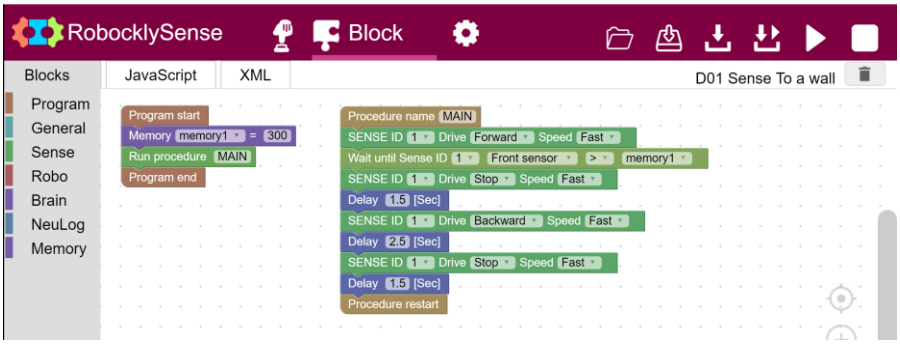
- Base unit
- 3 connectors for NeuLog sensors or brain units
- 5 IR range sensors
- A line sensor
- Pivot wheel
- 2 motors with wheels
- A controller that controls all the base sensors, motors, and independent operation
- A flash memory for the user programs

1.4 RobocklySense

The RobocklySense is a visual block-programming editor. It uses blocks that link together to build a program instead of writing code texts.

The RobocklySense uses special blocks that read the inputs, operating its outputs and read any of the NeuLog sensors.

The RobocklySense is very friendly and it is easy to create and run robotics programs.



1.5 RobocklySense installation

The software and drivers must be installed before connecting any modules to the PC.

1. Open the setup file on the CD you received with the system.
2. Follow the instructions on the screen. The installation process is straightforward and the required drivers are installed automatically.

The installation is composed of two parts: RobocklySense software installation and USB driver installation. After the installation process is completed, the RobocklySense software is ready to use.

The RobocklySense shortcut icon  should appear on the PC desktop.

Notes:
Upgrading the software can be done at any time. Installing the upgraded software just replaces the relevant files, so uninstalling the software before upgrading is not needed.

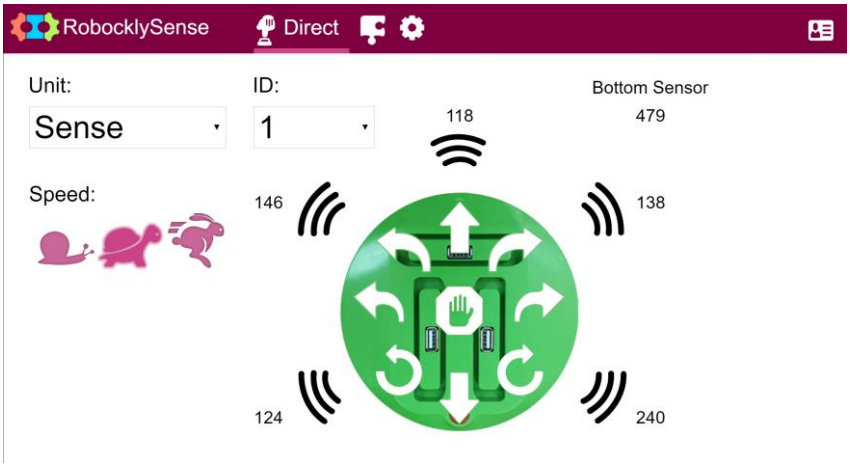
During upgrading the software the USB driver installation can be skipped by clicking the **Cancel** button.

1.6 Starting the RobocklySense program

You can find the RobocklySense program icon  on your computer desktop screen.

Click on the RobocklySense  icon to run the RobocklySense software.

The program is opened in a browser and the following screen appears:



This screen is for Direct mode (explained in experiment 1.1).

Exit is done in two steps.

1. Close the browser window.
2. Click on the RobocklySense  icon on the bottom and close the opened window.

Experiment 1.1 – Direct Mode

Objectives:

- To study the SENSE units and components.
- How to operate the SENSE units at direct mode.

Equipment required:

- Computer
- RobocklySense software
- SENSE autonomous

Discussion:

RobocklySense software enables to operate the SENSE controller directly.

We shall learn how to operate the SENSE motors and to read its sensors directly without programming.

Procedure:

1. Observe the SENSE.

2. Identify the communication cable inlet.

3. Hold the SENSE and turn it.

4. The SENSE includes:

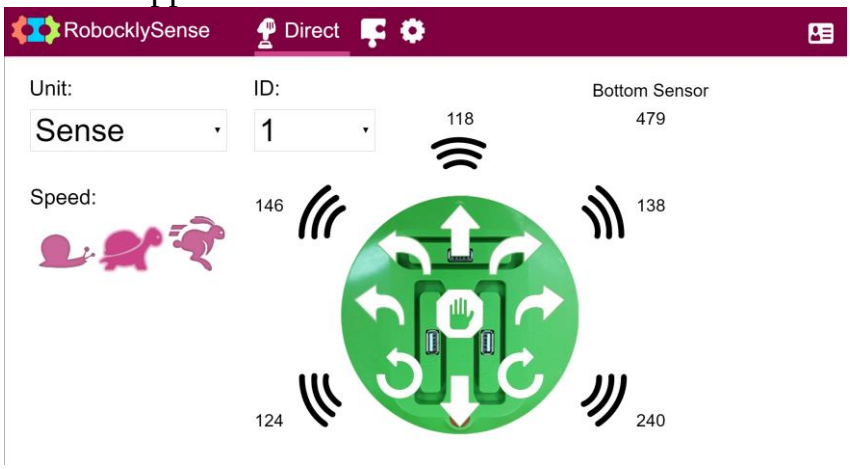
   - Two motors with wheels.
   - One pivot wheel.
   - Five range sensors made of Infra Red LED and phototransistor each.
   - A line (black or white) sensor made of Infra Red LED and phototransistor too.

   Identify them.

5. Connect the SENSE to the PC using the USB cable.

6. Run the RobocklySense software.

   The Direct screen appears:



The direct mode enables you to test the system and its wiring before programming and running the programs. This stage saves a lot of frustration in development.

This **Direct** mode window is for manual control and test of the robot optional units: **Sense, Robo, Robo Ex, Brain Arm, Brain Servo, Brain Motor** or **NeuLog sensor**.

The **Direct** screen is changed according to the selected unit. Each unit has its own default ID number. The user can change the module ID number. In this book, we shall refer to the default ID numbers of the units is 1.

1.1.1 Sense movement

The Sense robot has 9 movement commands:

Forward	both wheels rotate forward
Stop	both wheels stop
Backward	both wheels rotate backward
Left deviate	right wheel rotates forward fast and left wheel rotates forward slow
Left turn	right wheel rotates forward fast and left wheel stops
Left rotate	right wheel rotates forward fast and left wheel rotates backward fast
Right deviate	left wheel rotates forward fast and right wheel rotates forward slow
Right turn	left wheel rotates forward fast and right wheel stops
Right rotate	left wheel rotates forward fast and right wheel rotates backward fast

Each command has an arrow button on the **Direct** screen.

There are three buttons for changing the robot speed: 

7. Identify the arrow buttons.

8. Hold the SENSE in your hand.

9. Click on the buttons and observe the Sense robot reaction.

10. Place the Sense robot on your desk.

11. Click on the buttons and observe the Sense robot reaction.

1.1.2 The Sense sensors

The SENSE has five wall range sensors on its perimeter and one line sensor on its bottom, having the following names:

- Bottom sensor
- Front sensor
- Right front sensor
- Right back sensor
- Left front sensor
- Left back sensor

Each sensor is composed of Infra-Red (IR) LED (Light Emitting Diode) and phototransistor (light sensor) directed outside.

When the SENSE controller receives a command to read one of the range sensors, it lights its LED and measure the intensity of the received light.

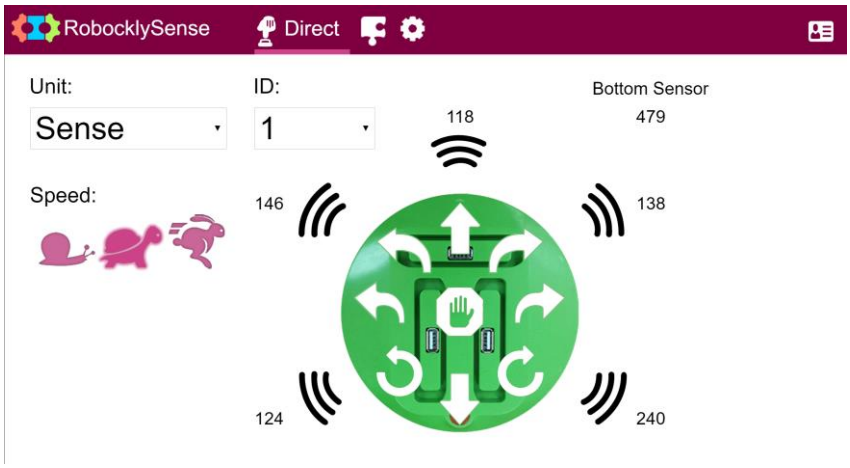
The range sensors are based on Infra-Red light in order to reduce the effect of the surrounding light.

A white surface returns much more light then a black surface. The line sensor on its bottom, based on this effect.

Note:
The range sensors are not calibrated. The read values represent the intensity of the received reflected IR light. For the same distance you may get different value from each sensor.

Pay attention to the side range sensors. They are all in angle of 45^{o}. The tracking a wall experiment (experiment 1.4) explains the reason for that.

12. Observe the **Direct** screen and the read values around the Sense picture. These values are the read values of the Sense sensors.



13. Place the Sense on a white surface.
The value of the bottom sensor should be above 500.

14. Place the Sense on a black surface.
The value of the bottom sensor should go down dramatically.

Note that there may be a big difference in the read value between different black surfaces.

15. Place the Sense robot in different distances from a wall and observe the effect on each of the five wall sensors.

Note
The read value of each sensor for the same distance may be different from sensor to sensor.

Experiment 1.2 – First Programs

Objectives:

- Using instructions for building a procedure.
- Downloading a program to the controller and running it.

Equipment required:

- Computer
- RobocklySense software
- SENSE Autonomous

Discussion:

A computer programs composed of chains of instructions according to the programming language instruction set. There are various programming languages with various instruction sets and various types programming.

We must tell the computer which instruction is the first instruction in the chain. The computer will execute this instruction and will continue to the following one in the chain.

The program may include instructions that move the computer to other instruction than the following one.

The program may include instructions that move the computer to other chains under condition or without condition. The following experiments describe these options.

There are many types of programming languages. Each one has its own syntax and its own set of instruction.

The RobocklySense is a visual block-programming editor. The RobocklySense uses blocks that link together to build a program and to concentrate on problem solving instead of writing code texts of a certain programming language.

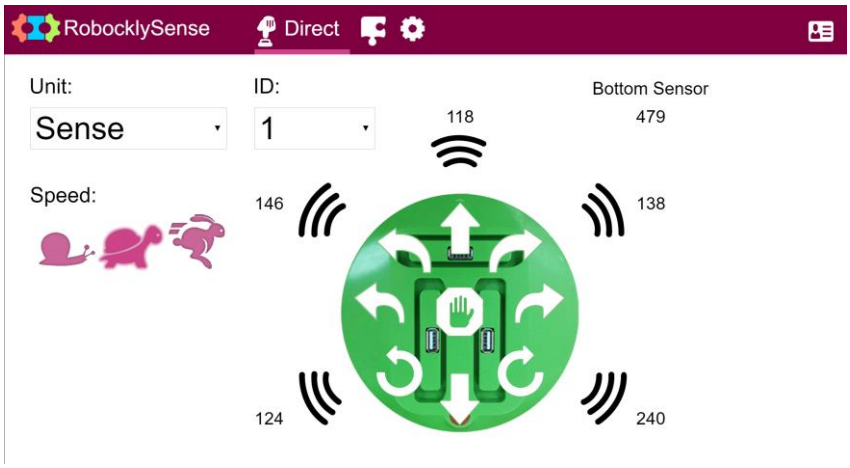
The RobocklySense is very friendly and it is easy to create and run robotics programs. It is powerful for robotic programs and the best software to start with.

Procedure:

1. Connect the SENSE to the PC using the USB cable.

2. Run the RobocklySense software.

   The Direct screen appears:

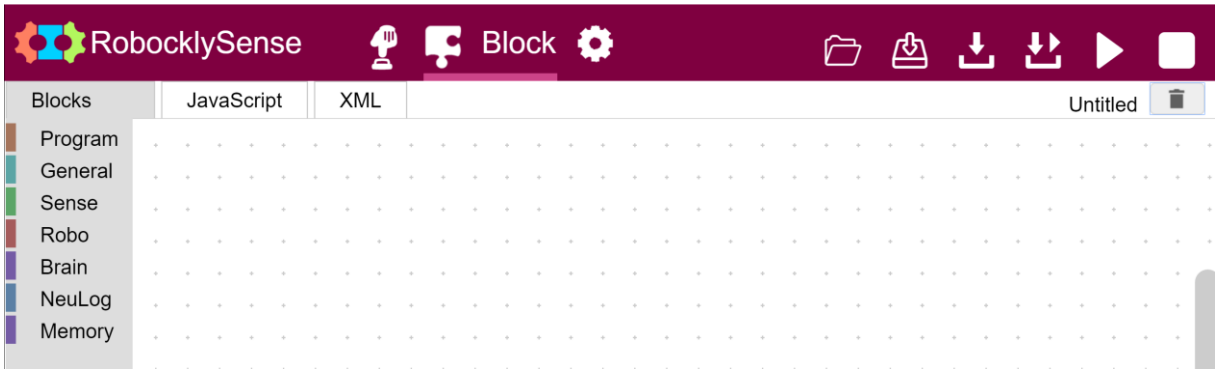


   The direct mode enables you to test the system and its wiring before programming and running the programs. This stage saves a lot of frustration in development.

3. Test the SENSE motors as described in experiment 1.1.

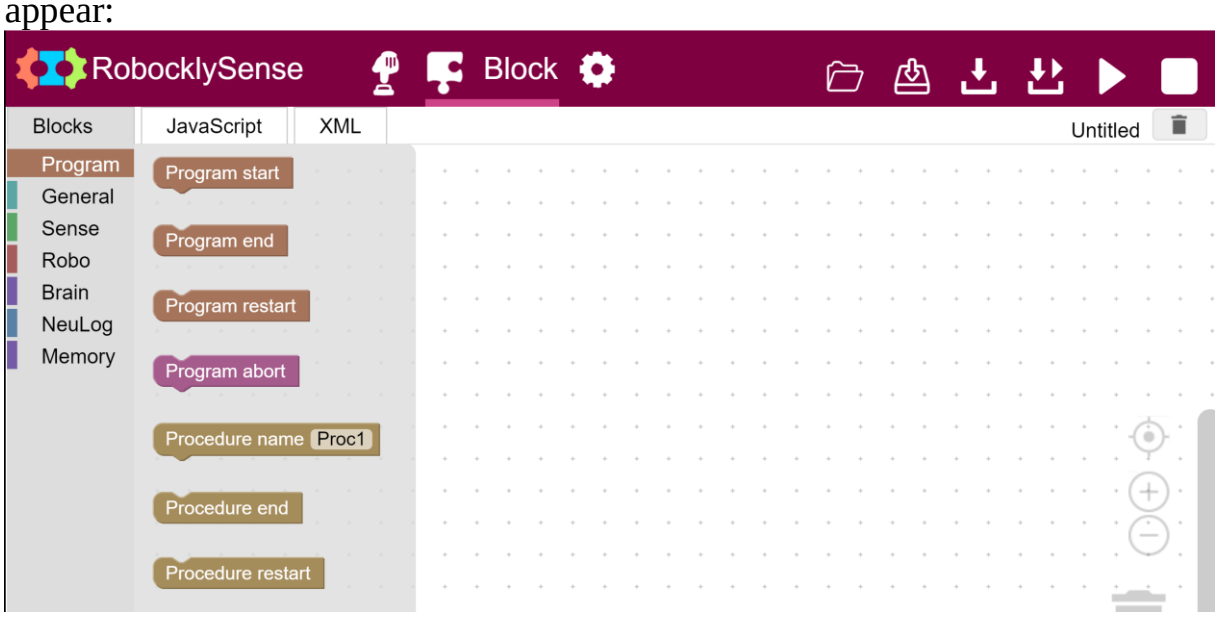
1.2.1 First program – forward

4. Move to **Block**  mode.

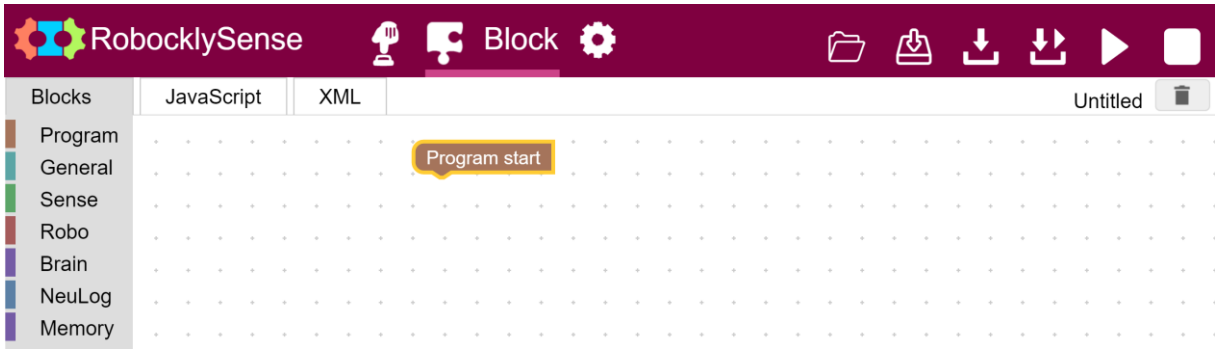


A computer program is composed of chains of instructions.

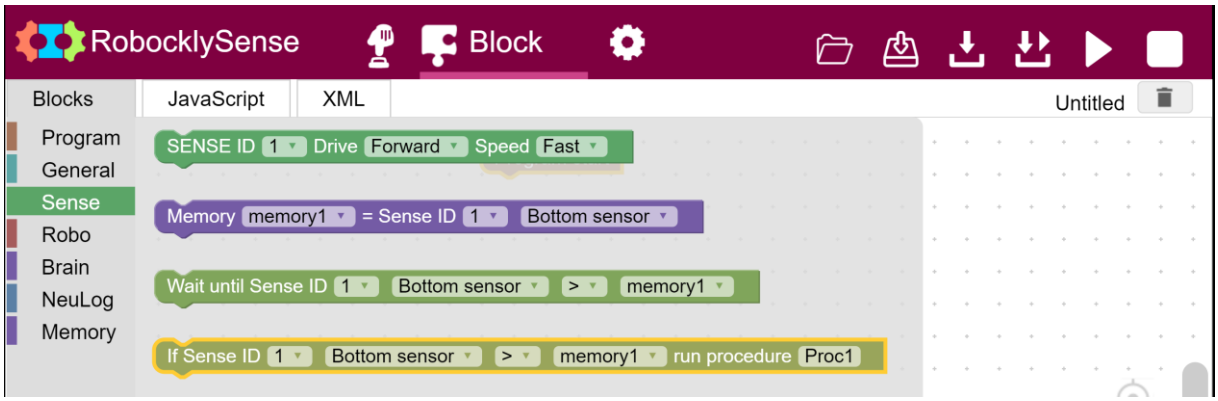
5. Click on the **Program** button and a list of program instruction list will appear:



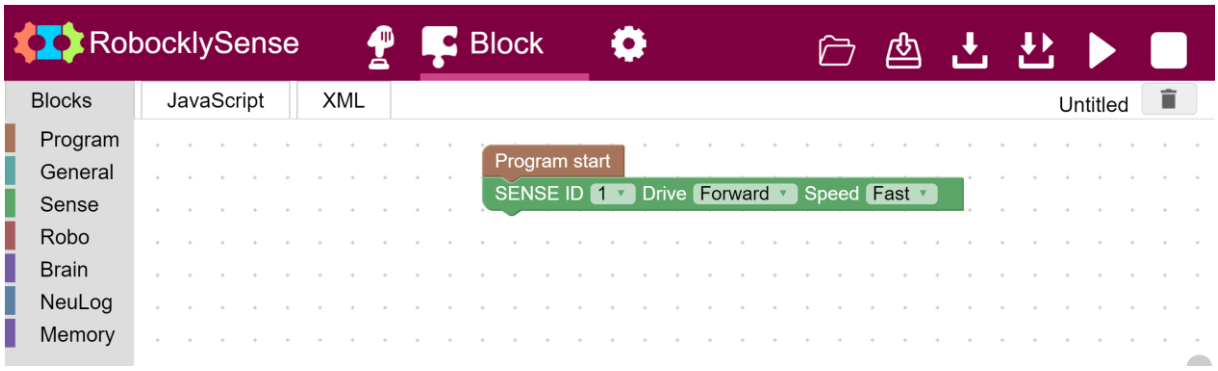
6. Click on the **Program start** instruction block and drag it to the right.



7. Click on the **Sense** button and a list of Sense instructions will appear:



8. Click on the **SENSE Drive** instruction block, drag it, and attach it to the **Program start** instruction block.
9.

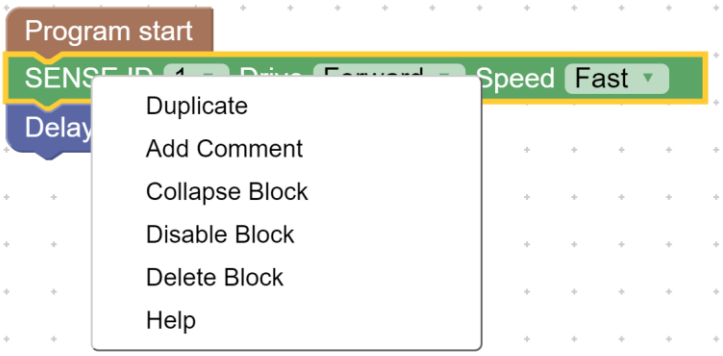


10. Click on the **General** button and select the **Delay** instruction block.

    Drag this block and attach it under the **Drive** instruction block.

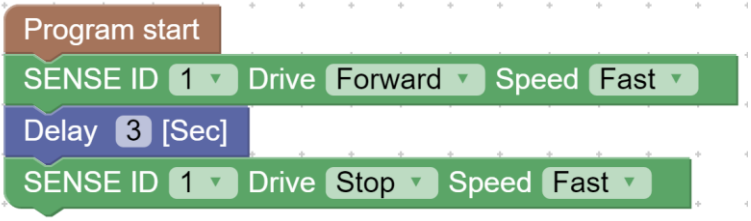
11. Change the delay value to 3 seconds.

12. Right click on the **Drive** instruction block.



13. Select **Duplicate** and a new **Drive** instruction block appear. Attach it to the **Delay** instruction block.

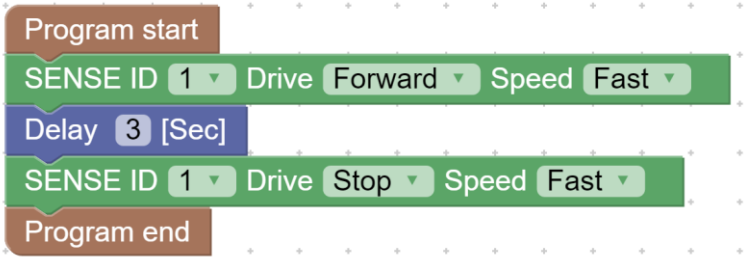
14. Change the **Drive** instruction to **Stop.**



15. Click on the **Program** button and select the **Program End** instruction block.

    Drag it under the **Drive Stop** instruction block.

16. Check that you have the following program:



17. Click on the **Save**  button and save the program under the name **PROG1**.

1.2.2 Program download

18. The SENSE controller is also a computer with a memory.

    We can download the program into its memory.

19. Click on the **Program Download**  button.

    This will transfer the PC program into the SENSE flash memory and replaces a previous program, if exists in it.

20. Click on the **Run**  button.

    The SENSE will move forward for 3 seconds and then stops.

    While running, the menu line is changed to the following with **Stop** button on the right.

21. To run the program in the SENSE memory, we can also use the Start/Stop orange pushbutton located on the SENSE panel.

    Press the SENSE panel pushbutton and you will see the SENSE move forward for 3 seconds and then stops.

    You may also have the Neulog battery module BAT-202.

22. The SENSE comes with an adapter for external battery. Such battery can be a standard Power Bank with USB outlet.

    When connecting such battery to the SENSE and disconnecting it from the PC, the robot becomes an independent robot running on its internal program in its flash memory.

    Connect the battery to the SENSE socket and disconnect the SENSE from the PC.

    The menu line is changed to the following:

23. Press the SENSE panel pushbutton and you will see the robot move forward for 3 seconds and then stops.

    The SENSE LED blinks while running.

24. Connect again the SENSE to the PC and wait until the menu line is changed back to the following.

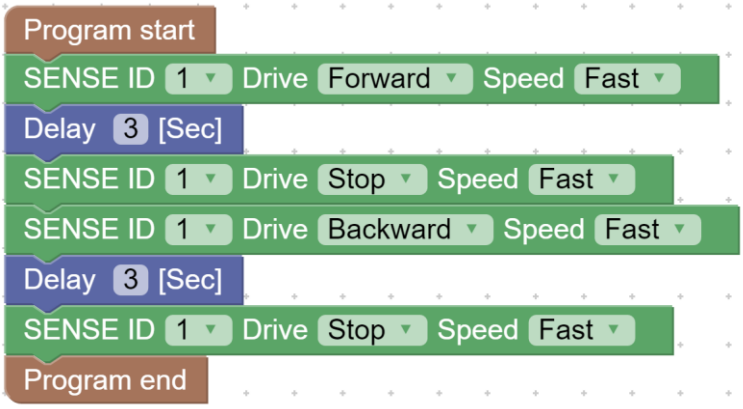
1.2.3 Forward and backward

25. Duplicate the **Delay** and the **Drive** instruction blocks.

26. Change the commands in the new **Drive** instruction block to **Backward**.

    It is better to stop a motor before changing its rotation direction.

27. Drag the instruction blocks and build the following program:



28. Click on the **Save**  button and save the program under the name **PROG2**.

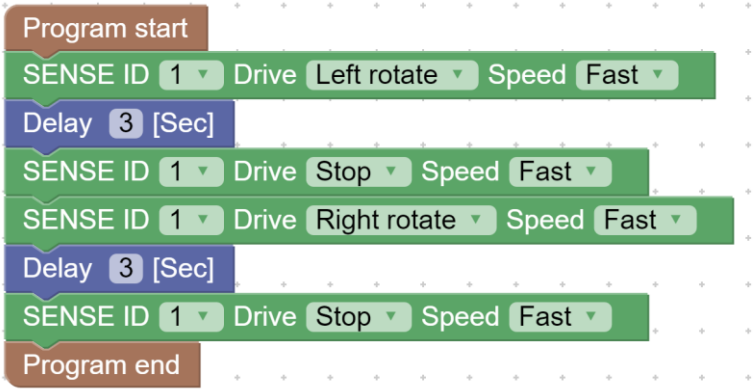
29. Click on the **Program Download**  button.

30. Click on the **Run**  button.

    The SENSE moves forward for 3 seconds, moves backward for 3 seconds and then stops.

1.2.4 Turning left and right

31. Change the program to the following program:



Pay attention to the **Drive** instructions.

32. Click on the **Save** button and save the program under the name **PROG3**.

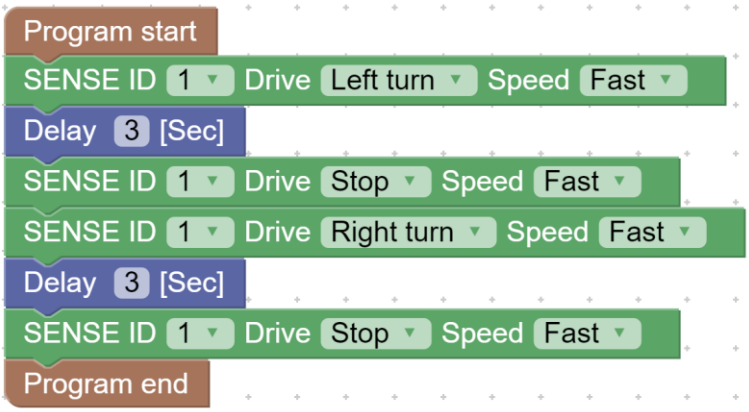
33. Click on the **Program Download** button.

34. Click on the **Run** button.

The SENSE rotates to the left for 3 seconds, rotates to the right for 3 seconds and then stops.

35. Change the delay time for having turns of 90°.

36. Change the program to the following program:



37. Click on the **Save**  button and save the program under the name **PROG4**.

38. Click on the **Program Download**  button.

39. Click on the **Run**  button.

    The SENSE rotates to the right for 3 seconds, rotates to the left for 3 seconds and then stops.

    What is the difference between the two rotating programs behavior?

40. Change the delay time for having turns of 90^{o}.

1.2.5 Challenge exercises

Task 1: Change the drive instructions in the program to the **Deviate** instructions.

Observe the SENSE movement.

Task 2: Make a program that moves the SENSE in a 30x30 cm square until it returns to its origin place.

Use the **Rotate** instructions for rotating.

Task 3: Make a program that moves the SENSE in a 30x30 cm square until it returns to its origin place.

Use the **Turn** instructions for rotating.

Experiment 1.3 – Interactive Programs

Objectives:

- Program that reacts to sensors.
- Using variables.
- Moving the SENSE between lines.
- Moving the SENSE between line and a wall.

Equipment required:

- Computer
- RobocklySense software
- SENSE

Discussion:

In this experiment, we will move the SENSE between two black lines. The position of the lines limits its motion. The robot changes direction when it finds a black line. This is an example of a system called a manipulator.

We will learn how to read and react to the Line and the Front Range sensors.

A closed loop system is a control system, which reacts to sensors and switches.

An automatic lighting system that includes light sensor is an example for closed loop system. The control system will light up the lamp when it is dark and turn it OFF when there is light. This system is automatically adapted to summer time (when the night is short) and to wintertime (when the night is long and starts early).

The program of closed loop system contains decision instructions such as:
'**Wait – until**', '**Stay while**', '**If – then**'.

The programs in this experiment use the '**Wait – until**' instruction.

1.3.1 Memories and Variables

In the delay instruction, we write a number that determines the length of the delay in seconds.

We call a number that is part of the instruction a **constant**. When we want to change this number, we have to search for the relevant instruction.

In programming, we prefer not to use constants but variables instead.

A **variable** is a memory cell with a name. In the instruction we indicate the name of the memory cell.

We can set the variable value in a certain place of the program, which saves us the searching for instructions.

The software includes special instruction for changing variable values according to program conditions.

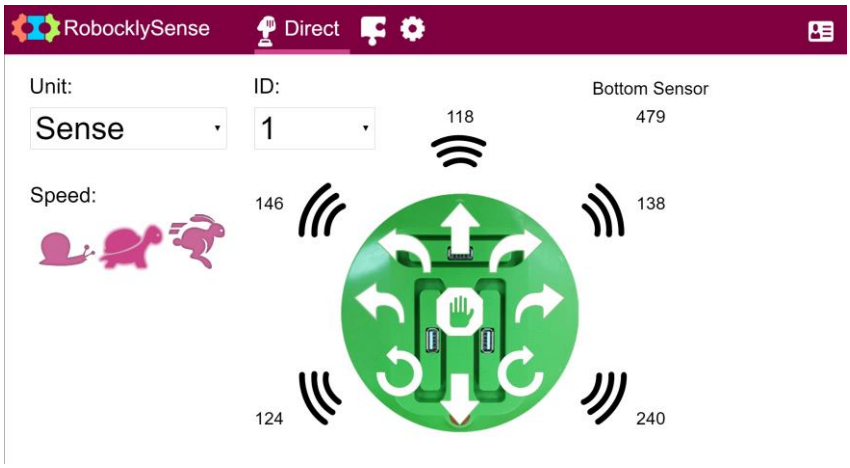
In **RobocklySense** we use memories (memoory1 to memory10) as variables.

Before starting the experiment, print two black lines as follows:

Procedure:

1. Connect the SENSE to the PC using the USB cable.

2. Run the RobocklySense software.



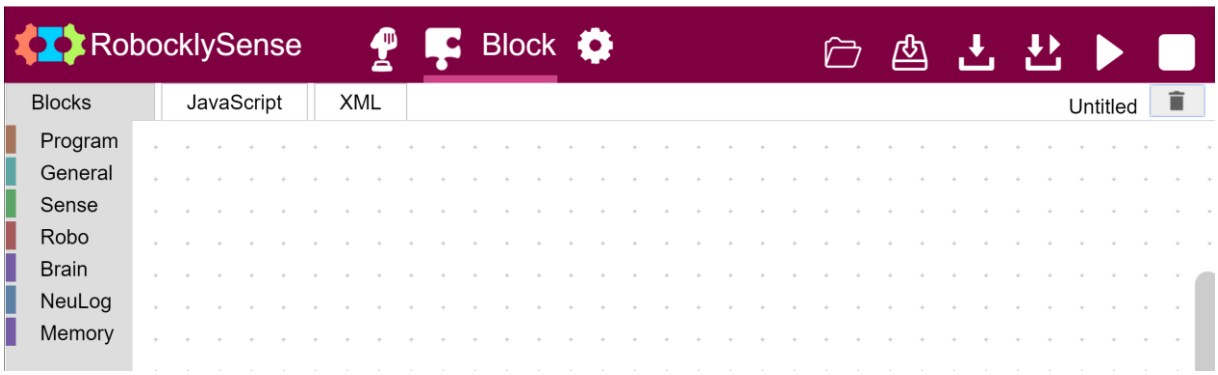
3. Test the SENSE motors as described in experiment 1.1.

4. Test the SENSE Bottom sensor as described in experiment 1.1.

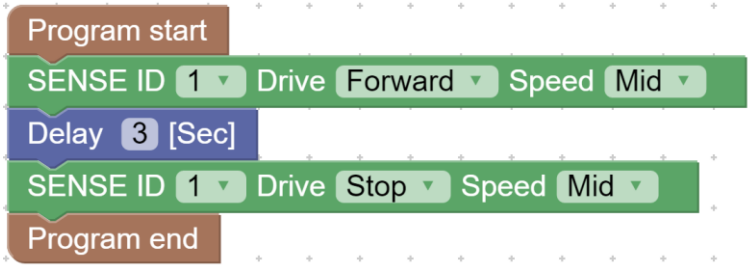
5. Record the values of the Bottom sensor when the SENSE is on a white surface. We shall call this value White Value.

6. Record the values of the Bottom sensor when the SENSE is on the black line. We shall call this value Black Value.

7. Move to **Block** mode.



8. Create the following program, which moves the SENSE forward for 3 seconds and stops.



9. Click on the **Save** button and save the program under the name **CART1**.

10. Click on the **Program Download** button.

11. Place the SENSE on the table or on the floor and click on the **Run** button.

    The SENSE will move forward for 3 seconds and then stops.

12. Disconnect the SENSE from the PC and plug the battery module.

13. Press the SENSE panel pushbutton and you will see the robot moves forward for 3 seconds and then stops.

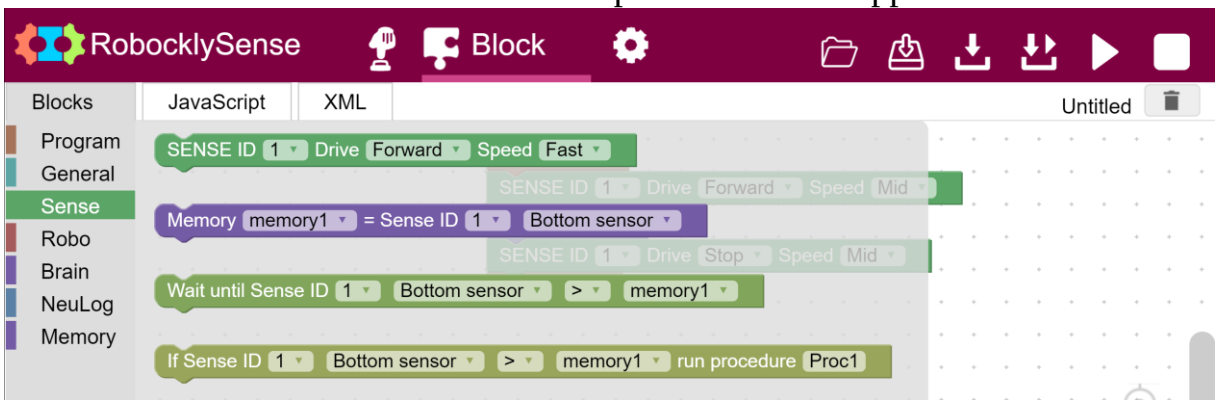
    The SENSE LED blinks while running.

14. Connect the SENSE to the PC.

1.3.2 Wait until

We shall replace the delay instruction with **Wait until** instruction.

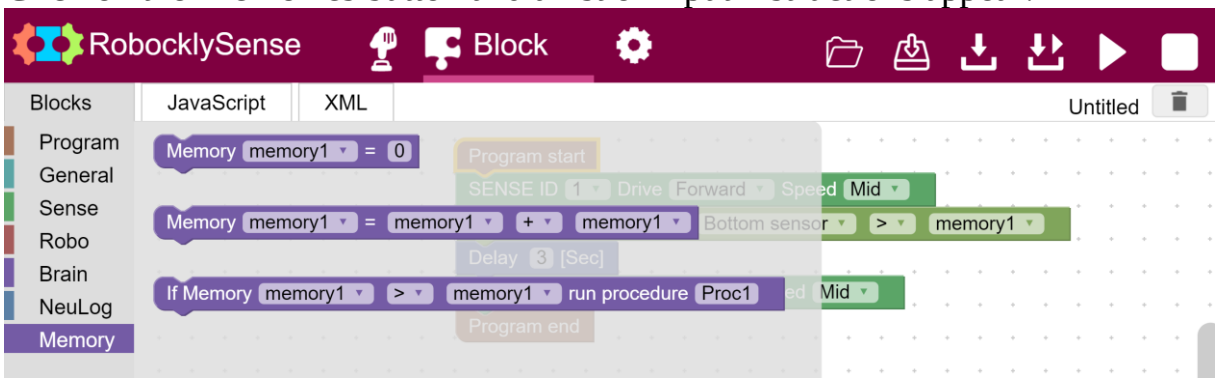
15. Click on the **Sense** button and a list of input instructions appear:



16. Click on the '**Wait until Sense ID 1 Bottom sensor > [memory1]**' instruction block and drag to the program above the **Delay** instruction block.

17. Change check sign from > (above) to < (below).

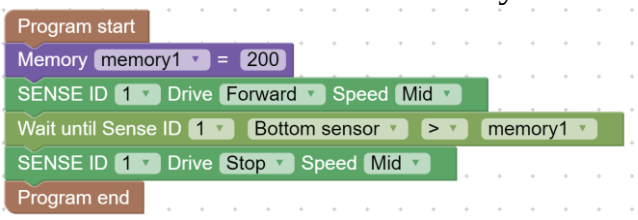
18. Click on the **Memories** button and a list of input instructions appear:



19. Click on the '**Memory [memory1] = 0**' instruction block and drag it to be the first instruction in the program.

20. Change the number in the instruction to 50 above the **Black Value**.

21. Delete the **Delay** instruction block and check that you have the following:



22. Place the SENSE on white surface with black line on it.

23. Run the program.

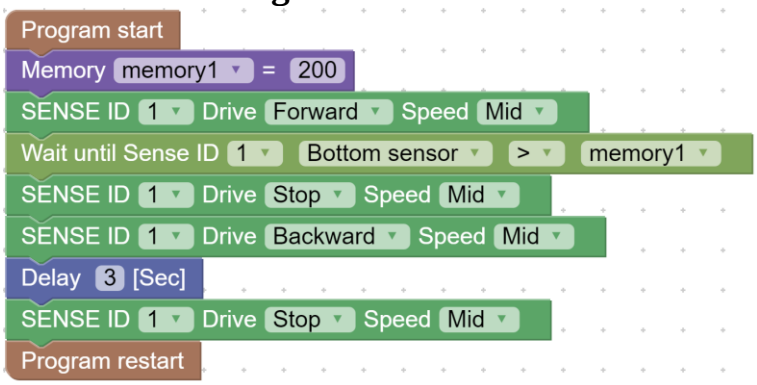
    The SENSE should move forward and stops when it reaches the black line.

1.3.3 Endless loop

Most of the control and robotic programs are programs that run in endless loop.

24. Change the program so that the SENSE goes forward and stops when it meets the black line, goes back for 3 seconds and forward again in endless loop.

25. Replace the **Program end** instruction block with **Program restart** instruction block from the **Program** instruction list.



26. Place the SENSE on white surface with black line on it.

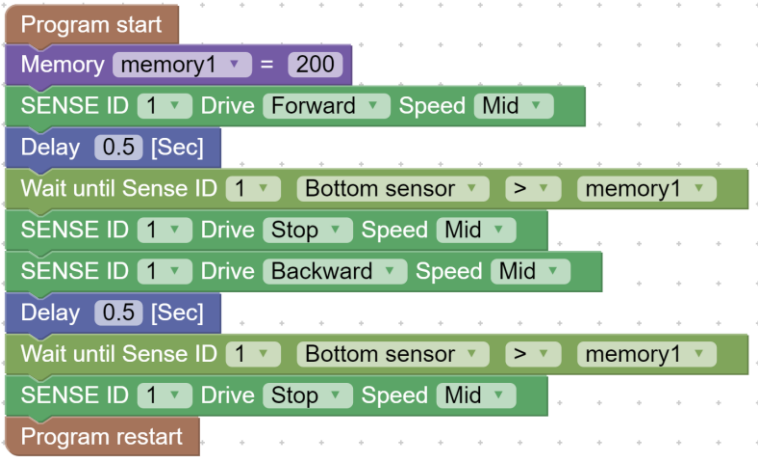
27. Run the program.

    The SENSE should move forward until it reaches the black line, stops, go backward for 3 seconds and forward again.

28. Stop the program by pressing the SENSE push-button or by clicking the **Stop** button, if the SENSE is connected to the computer.

1.3.4 Movement between two lines

29. Change the program so the SENSE will move between two lines.



Pay attention to the **Delay** instructions.

They are added in order to be sure that the robot will move from the current black line and wait before moving to the other black line.

30. Save the program under the name **CART2**.

31. Place the SENSE between two black lines and run the program.

The SENSE should run back and forth between the lines.

Increasing the distance between the lines changes the SENSE's travel accordingly.

1.3.5 Challenge exercise

Task 1: Improve the CART2 program so that the SENSE will move between a wall in front and a black line at the back.

Note:
You have to use the Front sensor while moving forward.
Take care for the compare sign (> or <).

Experiment 1.4 – Procedures as New Instructions

Objectives:

- Using procedures in a program.
- Reacting to sensors

Equipment required:

- Computer
- RobocklySense software
- SENSE Autonomous

Discussion:

A computer program composed of chains of instructions.

Instead of having a single chain of instructions, we can divide the program to procedures, which are short chains and give a name for each chain.

In a program, there is one main program and procedures. In this way, when we run the program, the computer knows where to start.

In this experiment, we shall build and use movement procedures.

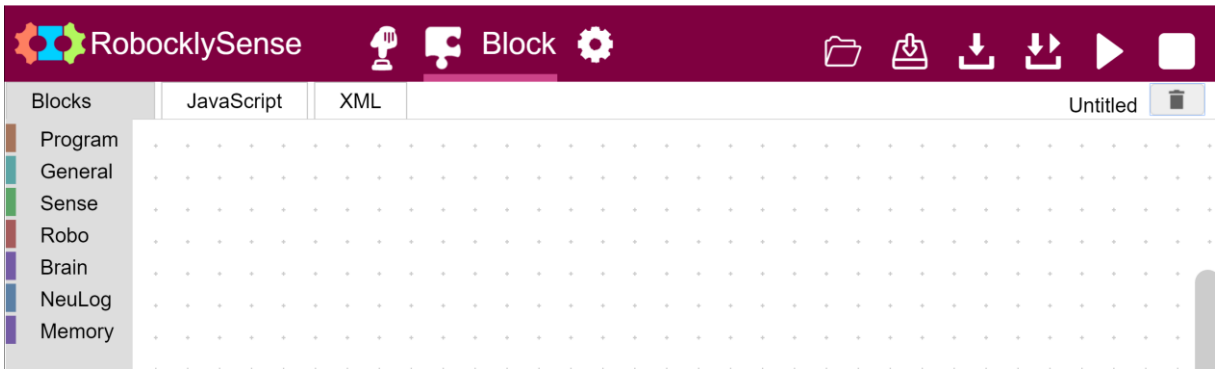
At the end of the program we need to add an instruction, which returns the program to the beginning (when we want the program to be executed repeatedly), or stops the program and returns to the operating system.

Procedure:

1. Connect the SENSE to the PC using the USB cable.

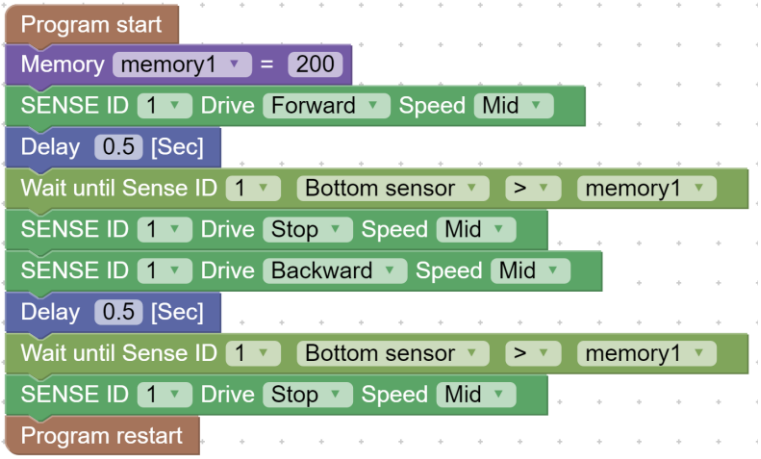
2. Run the RobocklySense software.

3. Move to **Block** mode.



4. Click on the **Open** button and open the program **CART2**.

5. Check that you have the following program:



   If not, build this program and save it under the name **CART2**.

6. Place the SENSE between two black lines and run the program.

   The SENSE should run back and forth between the lines.

   Increasing the distance between the lines changes the SENSE's travel accordingly.

7. Stop the program.

8. Click on the **Trash box** in order to clear the screen.

1.4.1 Programs and procedures

9. Click on the **Program** button and a list of general instructions appear.

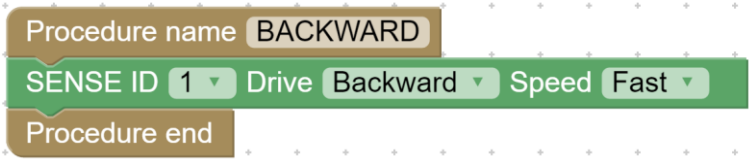
10. Click on the **Procedure name** button and drag it to the right.

11. The **Procedure name** instruction has a field for the procedure name. Write in this field the name **FORWARD**.

12. Create the following **FORWARD** procedure:

Note:
A procedure ends with **Procedure end** or **Procedure start**.

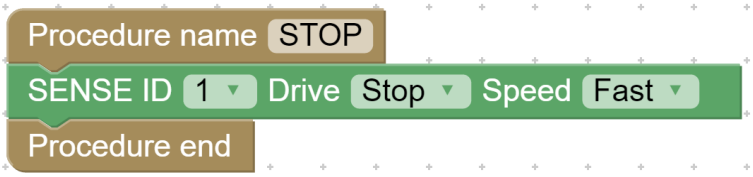
13. Create the following **BACKWARD** procedure:



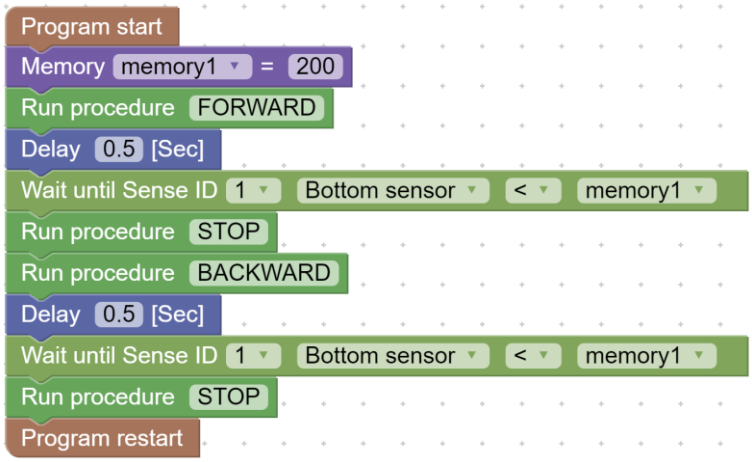
Note:
In order to duplicate a set of instruction, click on the first instruction, move it a little with the mouse, keep pressing the mouse button and click **ctrl+c**.

Now click **Ctrl+v** and this will duplicate the set of instructions.

14. Create the following **STOP** procedure:



15. Create the following main program:



Note:
Instead of using the **Drive** instructions in the main program, we use the **Run procedure** instruction from the **General** instruction list.

This opens us more options as we shall see later.

16. Save the program under the name **CART3**.

    The main program and the procedures are saved under this name.

    The main program makes the decisions and uses new instructions like forward, backward and stop.

17. The program function is the same as the function of the program **CART2**.

    Place the SENSE between two black lines and run the program.

    The SENSE should run back and forth between the lines.

    Increasing the distance between the lines changes the SENSE's travel accordingly.

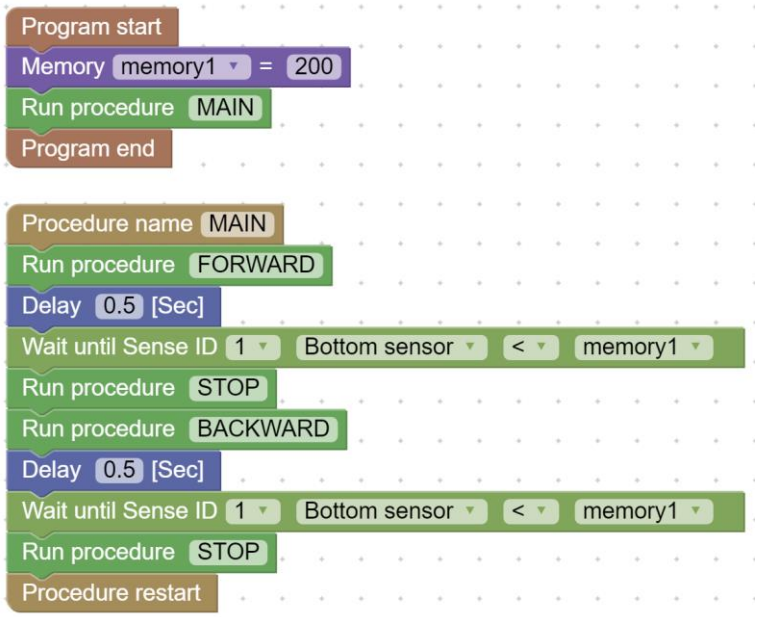
18. Stop the program.

1.4.2 Definitions

The definition of memory1 is done in every loop of the program although we have to do it only once. It also slow down the program cycle time.

In order to do it only once we create a main procedure without the definitions that runs in endless loop. The definitions are done in the program that calls after that the main procedure.

19. Change the program to the following program and MAIN procedure:



20. Save the program under the name **CART4**.

21. The program function is the same as the function of the program '**CART**3'.

    Place the SENSE between two black lines and run the program.

    The SENSE should run back and forth between the lines.

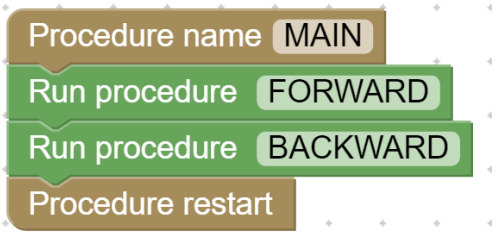
22. Stop the program.

1.4.3 Challenge exercises

Task 1: Improve the CART4 program so that the SENSE will move between a wall in front and a black line at the back.

Run and check.

Task 2: Improve the CART4 program so that the SENSE will move between a wall in front and a black line at the back, as in Task 1, but the main procedure should be as follows:



Change the FORWARD and the BACKWARD procedures accordingly.

1.4.4 Moving along a black line

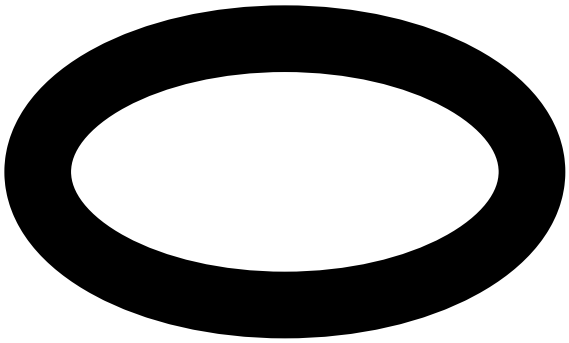
To move the SENSE along black line we use slow turn procedures of the SENSE.

In slow turns one wheel rotate and the other wheel stops. This way the SENSE still moves forward while turning.

In the main program, we do the movement according to the following idea:

Turning left until the SENSE find a black surface, and then turning right until the SENSE find a white surface.

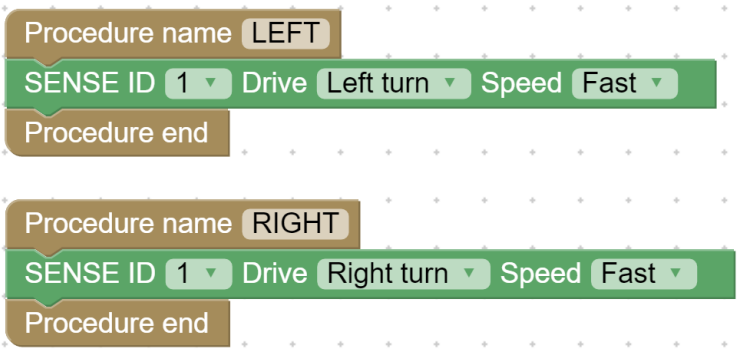
23. Print on a full page a black line as the following:



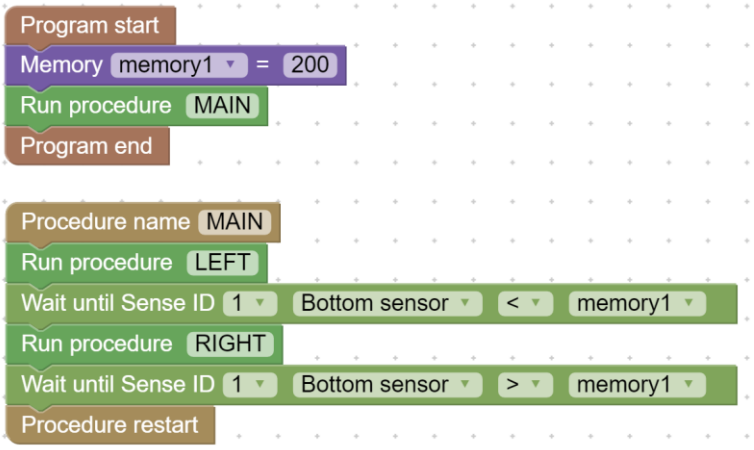
The width of the line should be at least 4cm.

24. Clear the screen.

25. Make **LEFT** and **RIGHT** turn procedures as follows:



26. Create the following main program and main procedure:



Note:
Pay attention to the compare signs (< and >).

27. Save the program under the name **CART5**.

28. Put the SENSE near the black line.

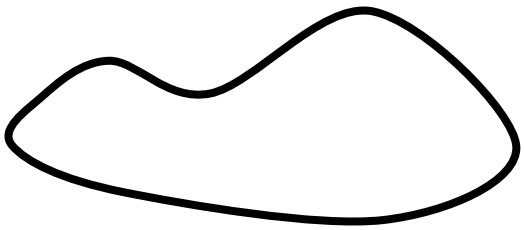
29. Run and check the SENSE movement.

30. Change the value of **memory1** to create smooth movement of the SENSE.

1.4.5 Challenge exercise

Task 1: Create different black lines for the SENSE and check its behavior. Improve the programs when needed.

Example of a line:



Experiment 1.5 – Conditions and Decisions

Objectives:

- The If instruction
- OR condition
- AND condition

Equipment required:

- Computer
- RobocklySense software
- SENSE Autonmous

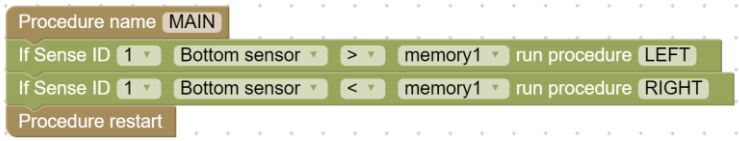
Discussion:

In the previous experiment, we learned about the **Wait until** instruction. This is one of the condition instructions.

The **If** instruction is the main condition instruction. It is composed of a condition and what procedure to operate when the condition exists.

The condition is checked when the condition instruction is executed. If the condition is not exists, the following instruction is executed.

The following program is a main procedure for moving along black line using **If** instructions.



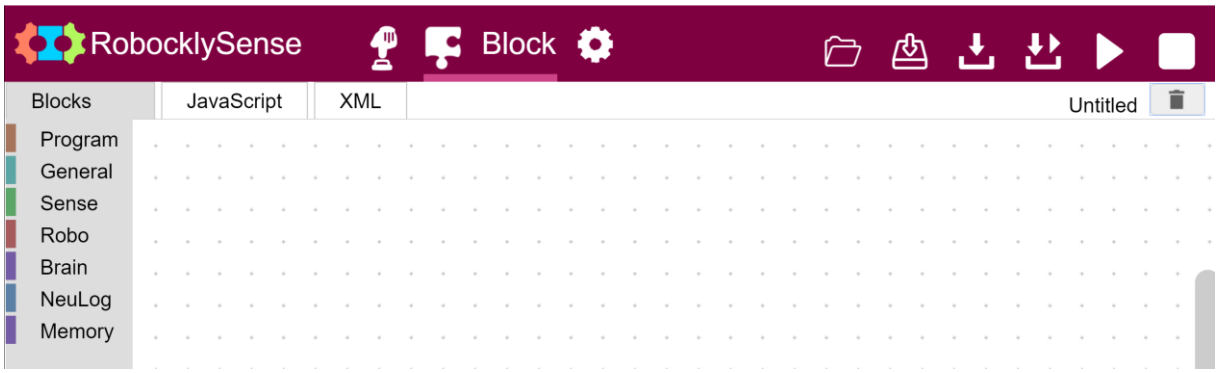
We can create complex conditions, called AND condition and OR condition, explained in the experiment later.

Procedure:

1. Connect the SENSE to the PC using the USB cable.

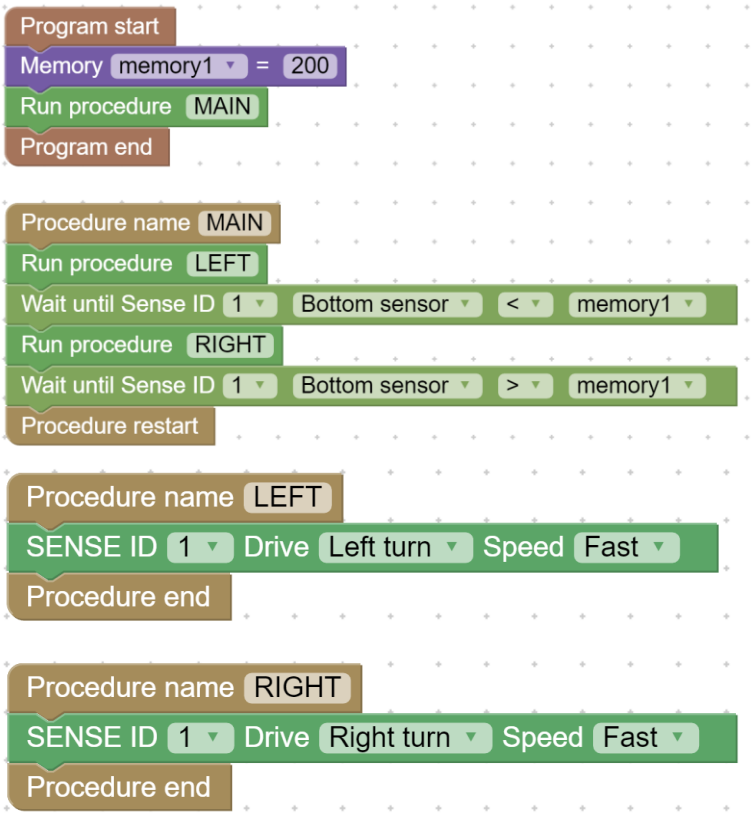
31. Run the RobocklySense software.

32. Move to **Block** mode.



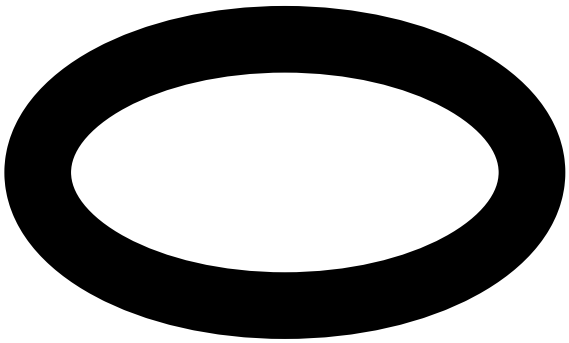
33. Click on the **Open** button and open the program **CART**5.

34. Check that you have the following program:



If not, build this program and save it under the name CART5.

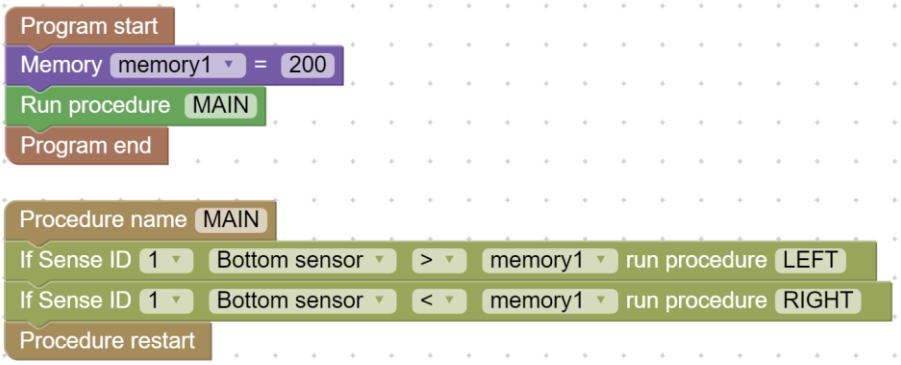
2. Place the SENSE on the black line circle and run the program.



The SENSE should go along the black line.

3. Stop the program.

4. Change the main program and the main procedure to the following:



The **If Sense** instruction is in the **Sense** instruction list.

5. Save the program under the name **CART6**.

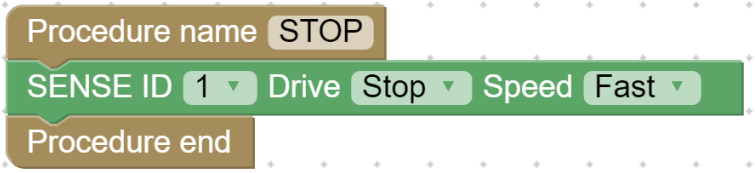
6. Put the SENSE near the black line.

7. Run and check the SENSE movement.

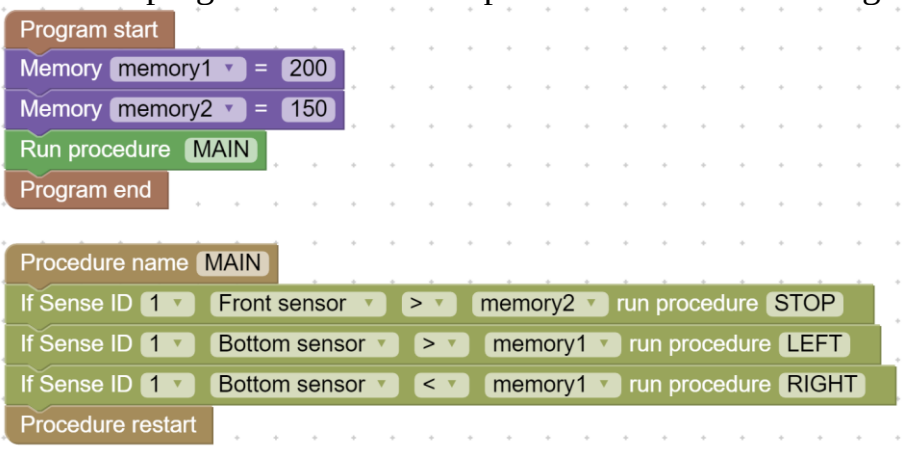
8. Change the value of memory1 to create smooth movement of the SENSE.

We shall improve the CART6 program so the SENSE stops, when you put your hand in front of it.

9. Build the following STOP procedure:



10. Change the main program and the main procedure to the following:



We added another variable that relates to the front wall sensor.

The main procedure checks in every cycle the distance from the wall and calls the STOP procedure when the SENSE is close to the wall.

11. Put the SENSE near the black line.

12. Run and check the SENSE movement.

13. Put your hand in front of the SENSE, while it moves.

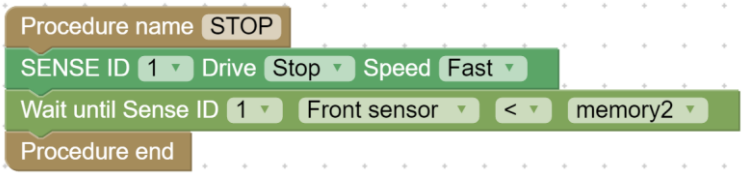
Instead of stopping the SENSE just slow down.

Think why.

14. We have to improve the STOP procedure to wait until you take off your hand.

    We prefer that the STOP procedure waits for a value below the value of stopping. We shall use the VAR4 for this **Wait until** instruction.

    Change the STOP procedure to the following:



15. Save the program under the name **CART7**.

16. Put the SENSE near the black line.

17. Run and check the SENSE movement.

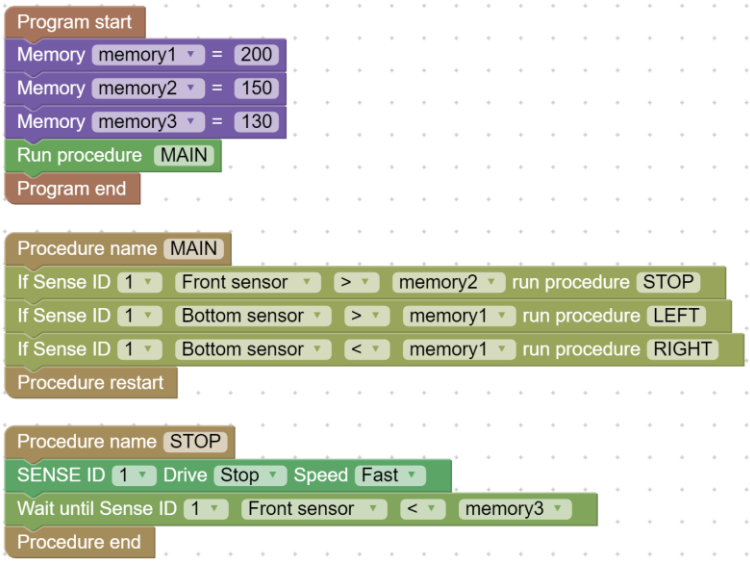
18. Put your hand in front of the SENSE, while it moves.

    Change the values of the memories until the SENSE works well.

1.5.1 OFF and ON with different values

In control systems, we usually prefer that the OFF condition value will be different from the ON condition value.
This in order to avoid bouncing of the system.

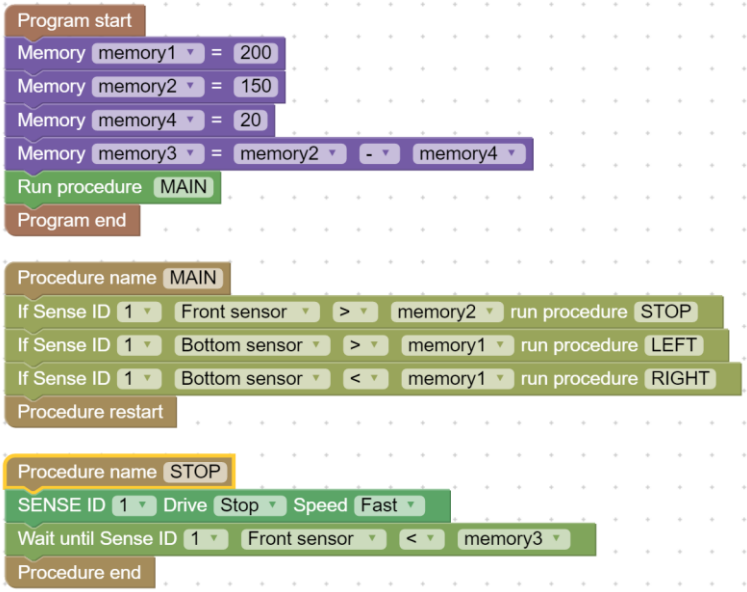
19. Change the program and procedures to the following:



20. The STOP value is higher than WAIT value.

21. Run and test this program.

22. When we change the STOP value, we have to change the WAIT value.
The following program takes care that the WAIT value (memory3) will be always lower than the STOP value (memory2).



23. Run and test this program.

1.5.2 AND condition

We would like to stop the SENSE only when it is close to the wall **and** only when it is on the black line.

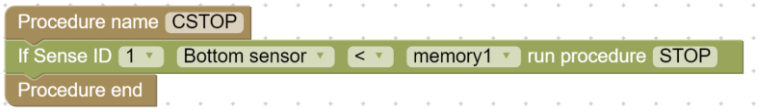
To achieve that we need the **AND** condition.

In some programming software, the **If** instruction can have two conditions with the **AND** condition between them.

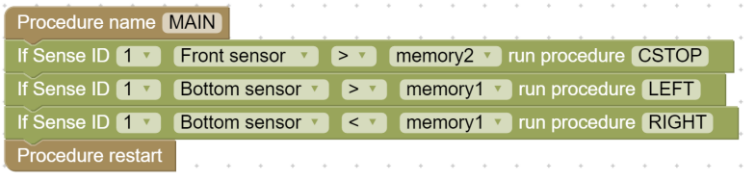
The RobocklySense does not have this option.

The AND operation can be achieved by creating another procedure.

24. Create the CSTOP (Conditional STOP) procedure with single instruction:



25. Change the main procedure to the following one:



26. Analyze the program and the procedures.

27. Save the program under the name **CART8**.

28. Put the SENSE near the black line.

29. Run and check the SENSE movement.

30. Put your hand in front of the SENSE, while it moves.

    Check the SENSE behavior.

1.5.3 OR condition

Instead of stopping near the wall, we would like that the SENSE will turn around and continue on the black line to the other direction.

The SENSE will turn to the left when it is white surface **or** when it is close to the wall **or** both.

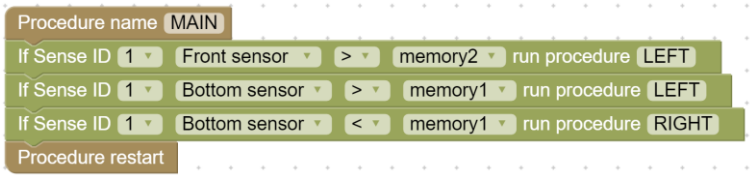
To achieve that we need the **OR** condition.

In some programming software, the **If** instruction can have two conditions with the **OR** condition between them.

The RobocklySense does not have this option.

The **OR** operation can be achieved by just writing the two **If** condition instruction one after the other.

31. Change the main procedure to the following one:



32. Analyze the program and the procedures.

33. Save the program under the name **CART9**.

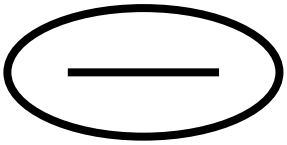
34. Put the SENSE near the black line.

35. Put a block on the black line.

36. Run and check the SENSE movement.

1.5.4 Challenge exercises

Task 1: Prepare two black lines in as the following:



Put obstacle on the inner line and let the SENSE move along this line.

When it meets the obstacle, it moves to the outer line and goes anlong it.

1.5.5 Movement along a wall

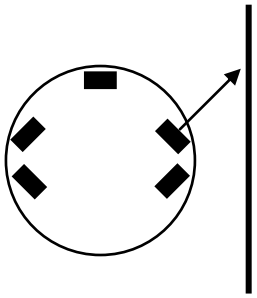
To move the SENSE along a wall we use the same algorithm of moving the SENSE along black line. We use the slow turn procedures.

In the main program, we use the **If** instruction to make the movement according to the following algorithm:

Turn left when the SENSE is too close to the wall.
Turn right when the SENSE is far from the wall.

To go along a wall on the right, we use the front side range-sensor.

The side range sensors are installed in 45° to the SENSE base.

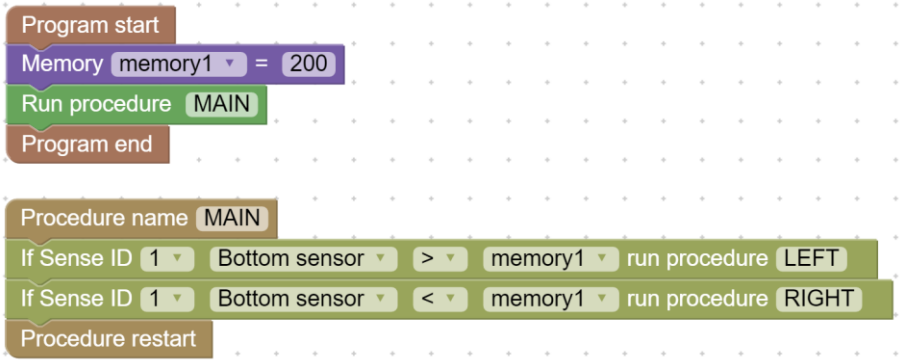


When the SENSE turns to the right, the measured distance is smaller than when it turns to the left.

Think what will happen if the range sensor is in parallel to the wall.

37. Take a round box (cylinder) as the first exercise wall.

38. **CART6** is the basic program for movement along a black line.

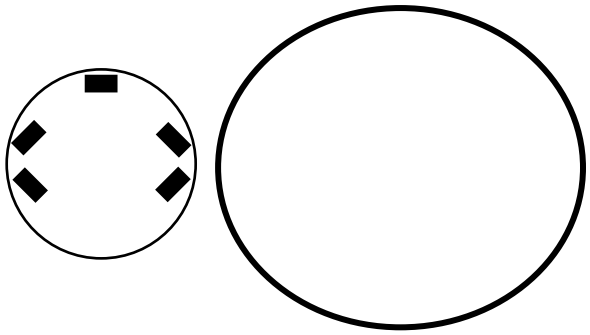


39. Change the main procedure of **CART6** to use the **Right Front sensor** instead of the **Bottom sensor**.

Change the value of **memory1** for distance of 4cm from the wall.

40. Save the program under the name **WALL1**.

41. Put the SENSE near the round box and run the program.



Check that the SENSE moves around the box.

1.5.6 Challenge exercises

Task 1: Improve the **WALL1** program so the SENSE goes forward when it does not sense a wall on its right side.

The SENSE stops when it meets a wall turn to the left and start moving along this wall.

Save this program under the name **WALL2**.

Task 2: Improve the **WALL2** program so the SENSE goes around a square box.

Put some obstacles on the SENSE way and improve the program so it will bypass them.

Chapter 2 – Brain Units

2.1 Brain units

Some of the input units can have their own "brain". The NeuLog sensors are such brain units. They send to the control unit, upon request, processed data such as: temperature (^{o}C or ^{o}F), light intensity in Lux, distance in meters, etc.

The output units can also be brain units. For example, unit that control the motor speed and direction, lamp intensity, servo motor angle, etc.

These brain units are connected in a chain to the main control unit, which communicates with them through messages.

2.2 NeuLog sensors as brain units

NeuLog sensors (Neuron Logger Sensors) are also brain units. Each sensor includes a tiny computer, which samples, processes and stores the sampled data. Each probe connected to the sensor is pre-calibrated in the factory and no further calibration is required.

The data provided by the sensor is processed digital data. The sensor includes different measurement ranges. Changing the measuring range or type of processing is done simply on the computer screen with NeuLog software.

The sensors are plugged to each other with almost no limitation on the composition and number of sensors in the chain.

NeuLog has over 50 different sensors. Some sensors perform as two to three sensors.

The SENSE has three sockets for NeuLog sensors.

Experiment 2.1 – Sound Sensor

Objectives:

- The sound sensor.
- Operating the SENSE by sound.

Equipment required:

- Computer
- RobocklySense software
- SENSE Autonomous
- NeuLog sound sensor

Discussion:

The sound sensor uses an internal microphone with a series of circuits, filters, and amplifiers to best isolate a specific sound source. Sound waves enter through the hole in the top of the sensor's plastic body so you should point that directly towards the sound source for best readings.

The sound sensor has two modes (ranges) of operation:

1. **Arbitrary analog units (Arb)** – An arbitrary unit demonstrates waves, frequencies, and periods. At this mode, the sound is sampled and reconstructed as a signal.

2. **Decibel (dB)** – A unit of measure to show the intensity (loudness of sound). Please note that this is a logarithmic unit.

   At this mode, the wave is sampled and the average intensity (calculated by the sensor controller) is converted into dB value. 40 dB represents silence.

In this experiment, we shall use it at dB mode and we assume its ID is 1 as the default ID.

Selecting the range should be done by the NeuLog software.

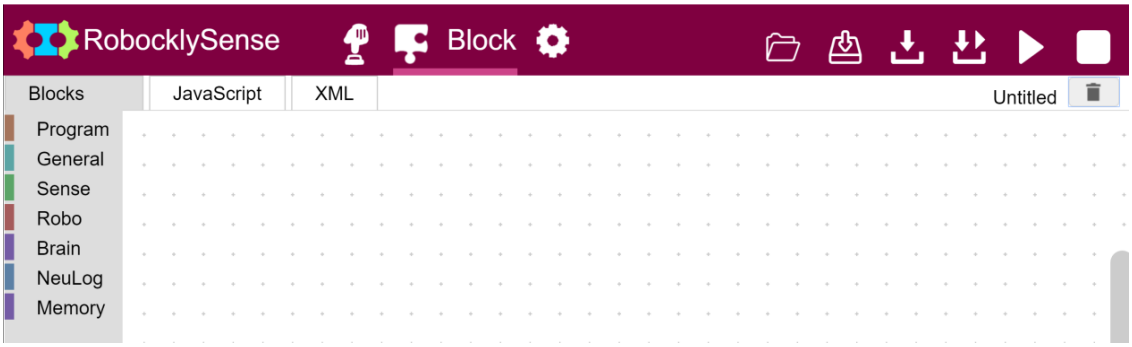
Procedure:

1. Plug the NeuLog sound sensor into one of the SENSE sockets.

2. Connect the SENSE to the PC using the USB cable.

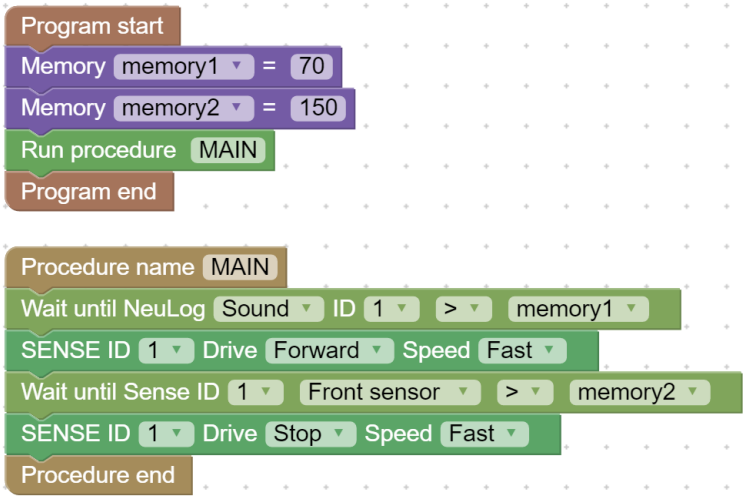
3. Run the **RobocklySense** software.

4. Move to **Block** mode.



We shall create a program that waits for sound of 70 dB, moves the SENSE to a wall, and stops.

5. Build the following program and the main procedure:



6. Observe the program and make sure that you understand all its instructions.

7. Place the SENSE in front of a wall and run the program.

   The SENSE should not move.

8. Clap your hand or make high sound.

   The SENSE should move to the wall and stop.

2.1.1 Challenge exercise

Task 1: Improve the program so:

(a) The SENSE will wait for a sound above 70 dB and then moves forward until meet a wall and stops.

(b) It will wait again for the sound, and moves backward until reaches a black line and stops.

(c) Return to the beginning.

Experiment 2.2 – Motion Sensor

Objectives:

- The motion sensor as distance sensor.
- Moving the robot according to the motion sensor.

Equipment required:

- Computer
- RobocklySense software
- SENSE Autonomous
- NeuLog motion sensor

Discussion:

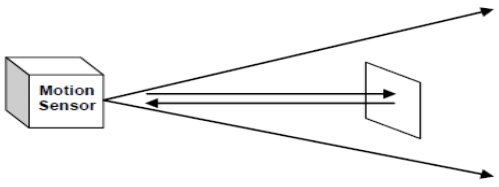
The motion sensor uses an ultrasonic transducer to both transmit an ultrasonic wave, and to measure its echo return. Objects in the range of 0.15 to 10 meters can accurately be measured to give distance, velocity, and acceleration readings using this method.

The motion sensor can collect data using the following measuring units:

- **Meters (m)** – The SI (International System of Units) distance unit
- **Meters/second (m/s)** – The SI velocity unit, which measures the distance traveled over time.
- **Meters/second^2 (m/s^2)** – The SI acceleration unit, which measures the change in velocity over time.

The motion sensor has two working ranges – one between 0.2 and 10.0 meters and one between 0.15 to 2 meters.

Ultrasonic waves that are emitted from the sensor and spread out in a cone pattern at about 15° around the point of reference.



The ultrasonic transducer is a device that can convert pulse train to transmitted ultrasonic pulses that can sense and convert back to electronic pulse train by another similar ultrasonic transducer or by itself.

The ultrasonic transducer is based on ceramic crystal, which is cut in a certain way and is placed between two metal plates. The crystal is characterized by the piezoelectric effect. Electrical field changes between the plates create mechanical vibrations in the crystal.

The crystal has a resonance frequency. The mechanical vibrations and electrical reactions depend on this resonance frequency.

Supplying pulses to the crystal of the ultrasonic transducer in a rate according to its frequency, causes it to vibrate and to transmit these pulses as an acoustic sound. This sound cannot be heard because it is above the hearing frequency range (usually it is at 40KHz).

The acoustic sound can be converted back to electronic pulses by another ultrasonic transducer or by the transmitter when it stops transmitting. The acoustic pulses vibrate this transducer and these vibrations are turned into voltage pulses.

The speed of the ultrasonic wave is about 300 m/s because it is a sound wave.

For distance measurement, a burst of the transducer frequency wave is sent and the system measures the time between the sending and the receiving.

$$\mathbf{S = 300 \cdot t}$$

Velocity is calculating by calculating the difference between two successive distances divided by the time between the samples (according to the sampling rate).

Acceleration is calculating by calculating the difference between two successive velocities divided by the time between the samples (according to the sampling rate).

The motion sensor uses a very sophisticated method that enables it to measure long distance range with a low power of pulses.

In this experiment, we shall use it at distance range and we assume its ID is 1 as the default ID. Selecting the range, should be done by the NeuLog software.

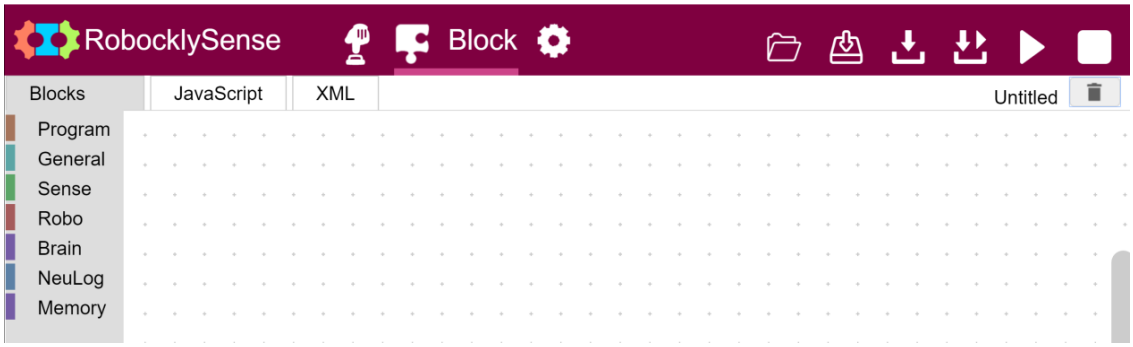
Procedure:

1. Plug the NeuLog motion sensor into one of the SENSE socket with its transducer directly to the front of the SENSE.

2. Connect the SENSE to the PC using the USB cable.

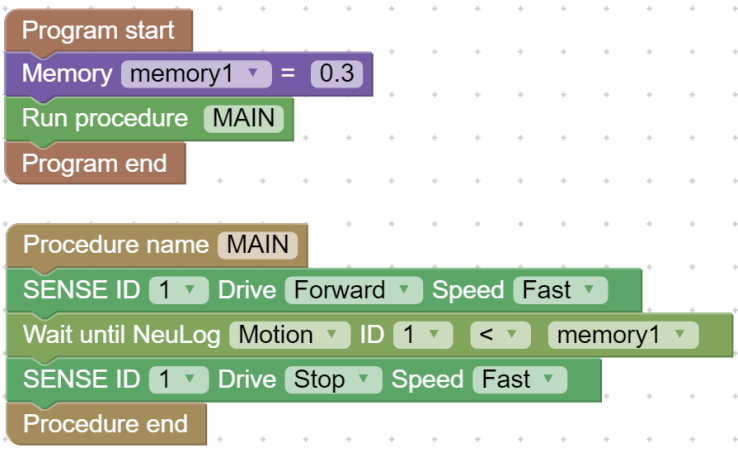
3. Run the **RobocklySense** software.

4. Move to **Block** mode.



We shall create a program that moves the SENSE forward to a wall and stops 30 cm in front of it.

5. Build the following program and the main procedure:



6. Observe the program and make sure that you understand all its instructions.

7. Place the SENSE in front of a wall and run the program.

The SENSE should move to the wall and stops 30 cm in front of it.

2.2.1 Challenge exercise

Going forward to a wall, stops 30cm before the wall, goes backward and stops at 50cm from the wall and return.

Task 1: Improve the program so the SENSE will:

- move to the wall,
- stops 30 cm in front of it,
- waits for 2 seconds,
- goes backward until a distance of 60 cm,
- stops for 2 second,
- returns to the beginning.

Experiment 2.3 – Brain Tracking Unit

Objectives:

- The brain tracking unit.
- Moving to an IR (Infra Red) transmitter.
- Following an IR transmitter.

Equipment required:

- Computer
- RobocklySense software
- SENSE Autonomous
- Brain tracking unit
- IR transmitter

Discussion:

2.3.1 IR Transmitter

The infra red transmitter can be plugged into any of the SENSE sockets or in the backup battery socket to be followed by the brain tracking unit.

Infrared light is light that transmitted from a heat source. We cannot see the IR light. The frequency of this light is a little below the red light and this is why we call it infra (before) red.

The surrounding light does not affect this light much.

2.3.2 Brain tracking unit

The brain unit, in a rigid plastic case, can be plugged into one of the SENSE sockets.

The brain tracking unit has three IR (Infra Red) sensors that enables to track the IR transmitter.

Two IR sensors are at the same line with opaque partition between them.

When IR light falls on both of them, it means that the SENSE is in front of the IR light source.

When the SENSE is at angle to the light source, the IR light will fall only on one of the IR sensors.

The third IR sensor measures the environment IR light. The brain unit controller uses this measurement to eliminate the environment light.

The brain unit output is a number (based on binary number) that describes the detection status of an IR transmitter as the following:

0 (00) – No IR transmitter light
1 (01) – IR transmitter light on the right
2 (10) – IR transmitter light on the left
3 (11) – IR transmitter light at front

Procedure:

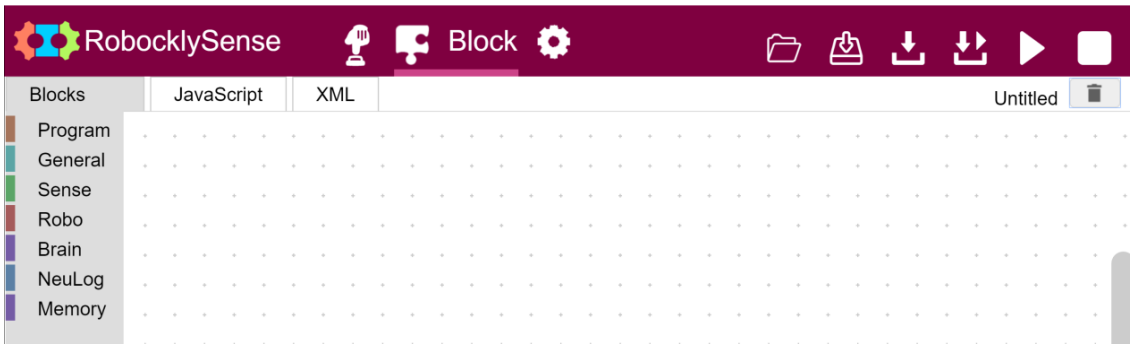
1. Plug the brain tracking unit into the front socket of the SENSE.

2. Plug the IR transmitter into a backup battery. Put the battery backup on a 3cm high surface.

3. Connect the SENSE to the PC using the USB cable.

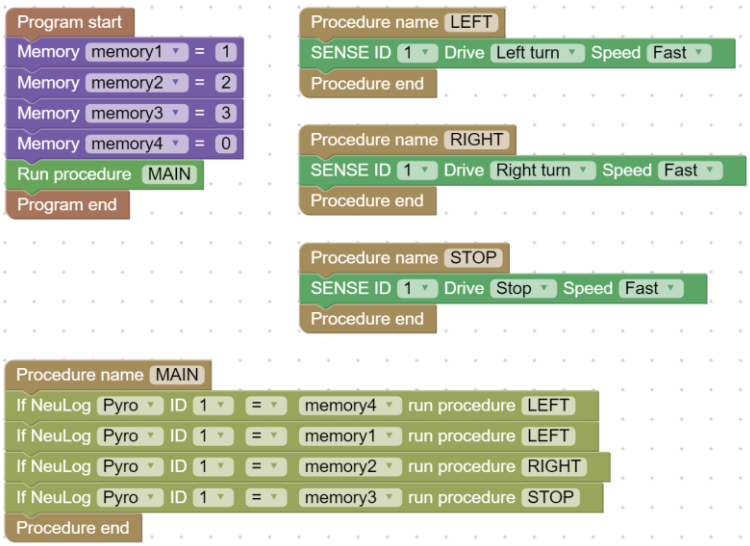
4. Run the RobocklySense software.

5. Move to **Block** mode.



We shall create a program that turns the SENSE to the left until it 'sees' the IR transmitter and then tracks it without moving (just rotates).

6. Build the following program and the main procedure:



7. Observe the program and make sure that you understand all its instructions.

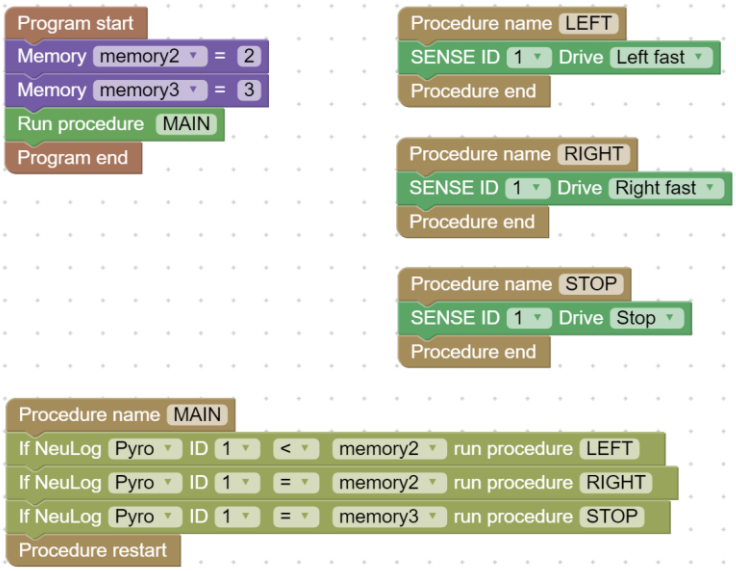
8. Download the program.

9. Place the SENSE on the floor and run the program.

   The SENSE should turn to the left until it 'sees' the infrared beam.

10. Move the IR transmitter slowly and check that the SENSE tracks it.

11. Change the program to the following:



12. This program should work the same as the previous one. Check that.

2.3.3 Challenge exercise

Task 1: Build program and procedures so the SENSE will:

- turns around until it finds an IR light,
- moves to IR transmitter,
- stops in front of it.