# Sense Autonomous

NeuLog

# Sense Autonomous

2_28



For further information contact info@neulog.com

# 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 electronics. A few years ago, we did not see terms such as wearable computing or internet of things.

Everyday a surprising new product or application appears and months later, we cannot realize how we lived without it. Modern systems are based more and more on machine learning and artificial intelligence.

The robotic systems which are part of the embedded computer system, perform independent activities such as: search, manipulation, identification, activation, protection etc.

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

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

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.

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 autonomous is a robot enables us to program many robot applications and functions such as movement on a line, movement along walls, tracking, AGV (Automatic Guided Vehicle), autonomic car, autonomic guard vehicle, autonomic taxi driver, environment monitoring, manipulating car and more. All these applications are described as exercises in this book.

# 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 multiple variables, which govern the behavior of the system.

**Examples:**

- Air-conditioning systems control the temperature in the room.
- 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 requirements.

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 as the results of the inputs and the program "decisions".

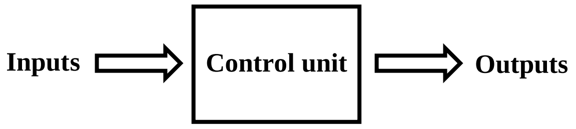



**Figure 1-1**

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

Disconnecting the control unit from the computer and connecting a power source such as a battery to it will create 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.
- **Autonomous vehicle** such as: AGV, autonomous car, autonomous guard vehicle, autonomous taxi driver, autonomous manipulator.
- Following a moving body holding IR transmitter using tracking module.
- **Environmental monitoring** and measurement robot with NeuLog sensors.

The **SENSE autonomous** has the following built in:

- Base unit
- 3 connectors for NeuLog sensors or brain units
- 5 IR range sensors
- 1 line sensor
- Pivot wheel
- 2 motors with wheels
- A controller for the base sensors, motors, and independent operation
- A flash memory for the user programs
- USB connector for connection to PC or MAC

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

You may also have the NeuLog battery module BAT-202, which can be plugged directly into one of the SENSE sockets.

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

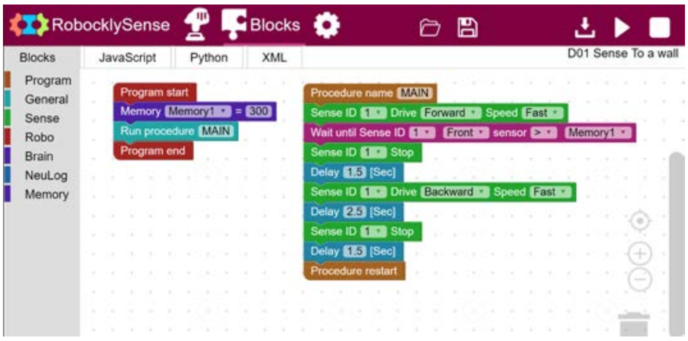
In this book, we shall call the sense autonomous in short SENSE.

## 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 can read the inputs, operate the outputs, and read any of the data from NeuLog sensors.

The RobocklySense is very user 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 or MAC.**

1. Download **RobocklySense Application** from: **www.neulog.com**.

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 necessary.

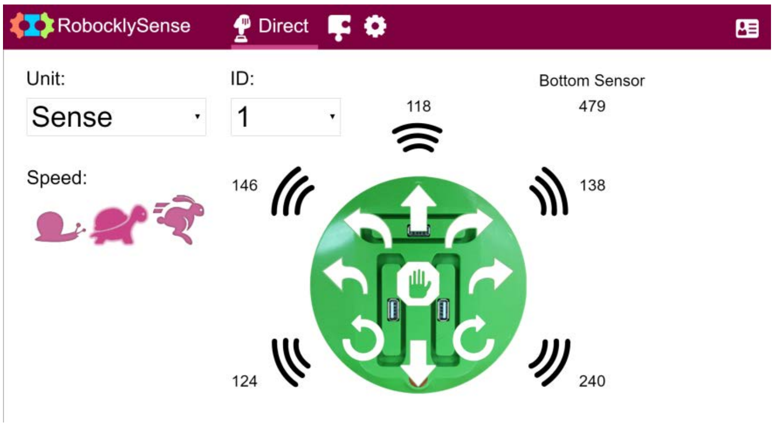
During an upgrade, the software the USB driver installation can be skipped by clicking the **Cancel** button.

## 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 in direct mode.

## Equipment required:

- Computer
- RobocklySense software
- SENSE autonomous
- USB connection cable

## Discussion:

RobocklySense software enables the user to operate the SENSE controller directly.

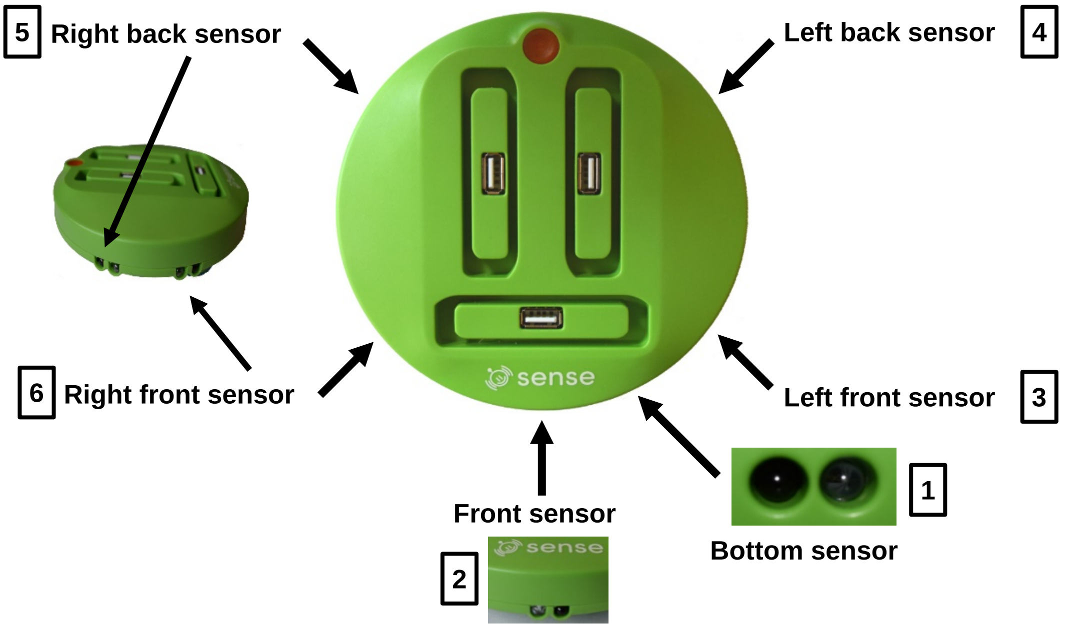
We shall learn how to operate the SENSE motors and to read its sensors directly without programming.

# 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 an infrared (IR) LED and phototransistor each.
   - A line (black or white) sensor made of an infrared LED and phototransistor too.

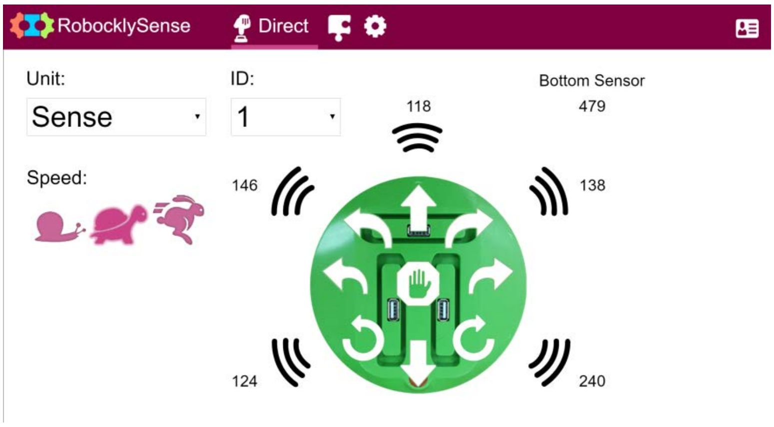
   Identify them.




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 the sensors' readings 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 following screens are **Direct** mode windows of the **Robo, Brain arm and NeuLog sensor.**




Brain arm and NeuLog sensor are explained in chapter 3.

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 number of the unit as 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 slower a little |
| **Left turn** | right wheel rotates forward fast and left wheel rotates forward very slowly |
| **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 slower a little |
| **Right turn** | left wheel rotates forward fast and right wheel rotates forward very slowly |
| **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: 

1. Identify the arrow buttons.

2. Hold the SENSE in your hand.

3. Click on the buttons and observe the SENSE robot reaction.

4. Place the Sense robot on your desk.

5. Again, 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 an infrared (IR) LED (Light Emitting Diode) and phototransistor (light sensor) directed outward.

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

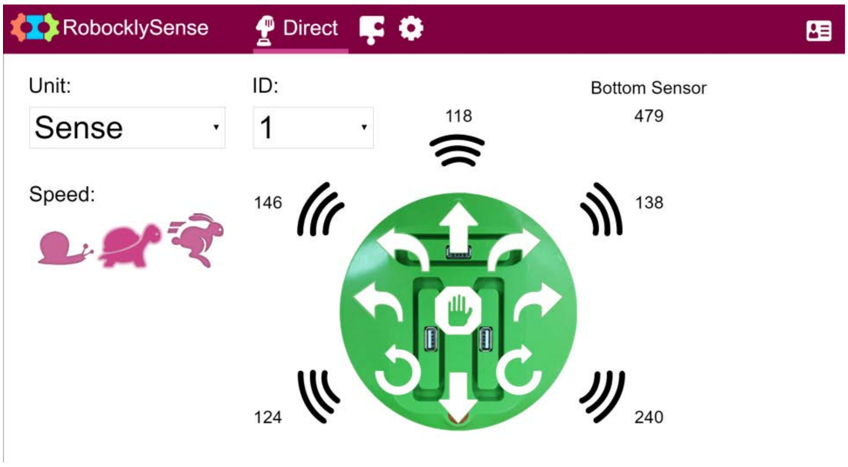
The range sensors are based on infrared light in order to reduce the effect of the surrounding light.

A white surface returns much more light then a black surface. Also colored surfaces return different values. The line sensor on the bottom is 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 at an angle of 45º. The Tracking a Wall experiment (experiment 1.4) explains the reason for that.

1. Observe the **Direct** screen and the read values around the SENSE picture.




These values are the read values of the SENSE sensors.

2. Place the SENSE on a white surface.

   The value of the bottom sensor should be above 500.

3. Place the SENSE on a black surface.

   The value of the bottom sensor should go down dramatically.

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

4. Place the SENSE robot at 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
- USB connection cable
- BAT-202 Battery module

## Discussion:

A computer program is composed of chains of instructions according to the programming language instruction set. There are various programming languages, with different instruction sets and multiple types of programming.

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

The program may include instructions that direct the computer to other instructions other than the subsequent one.

The program may include instructions that move the computer to other chains under condition.

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 in a certain programming language.

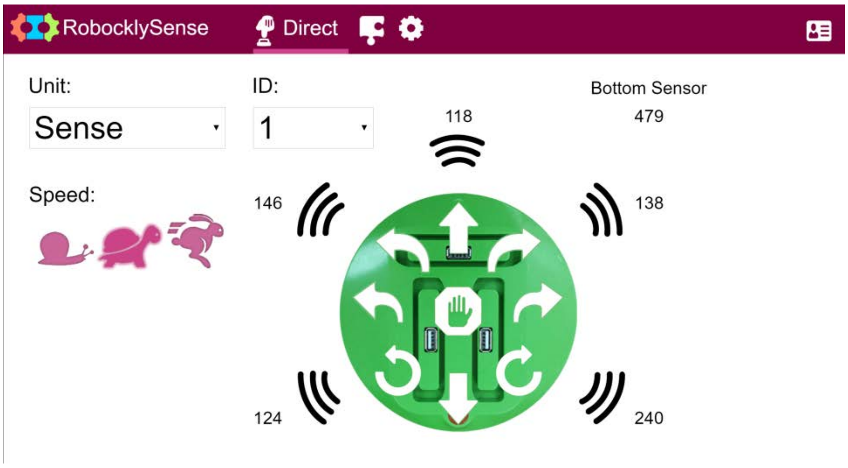
The RobocklySense is very user friendly and it is easy to create and run robotics programs. It is powerful for robotic programs and the best software to start with.

# Procedure:

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

2. Run the **RobocklySense** software.

   The Direct screen appears:

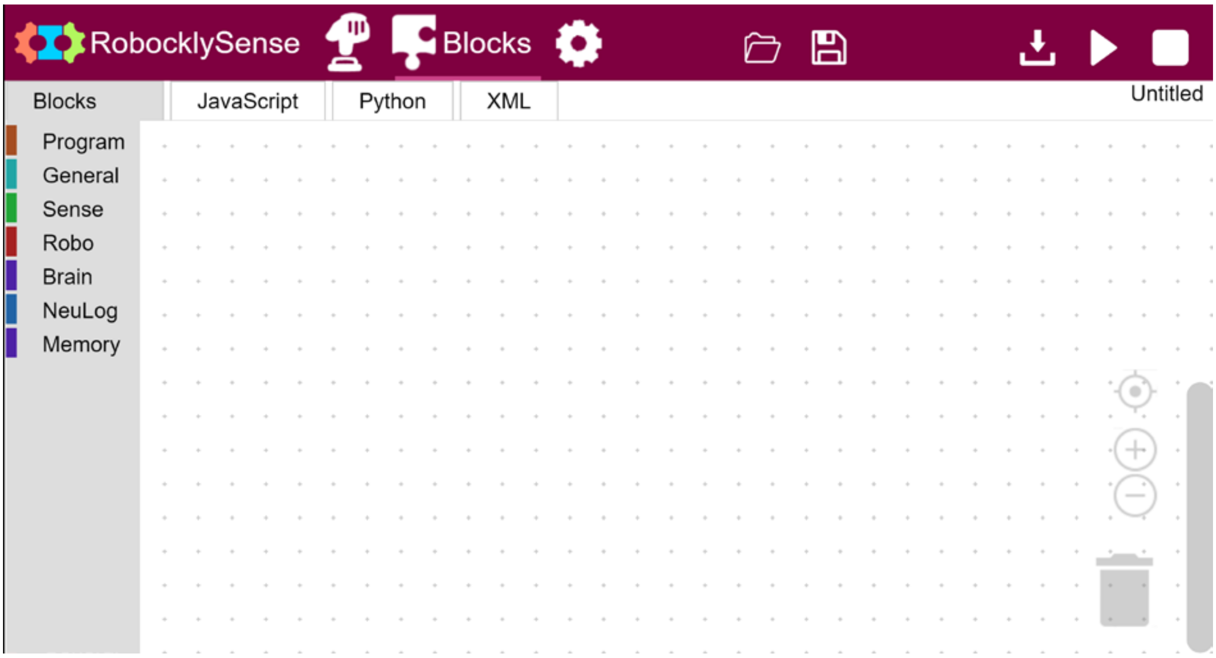



   The direct mode enables you to test the system and its wiring before programming and running the programs. This stage saves a lot of frustration in development.

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

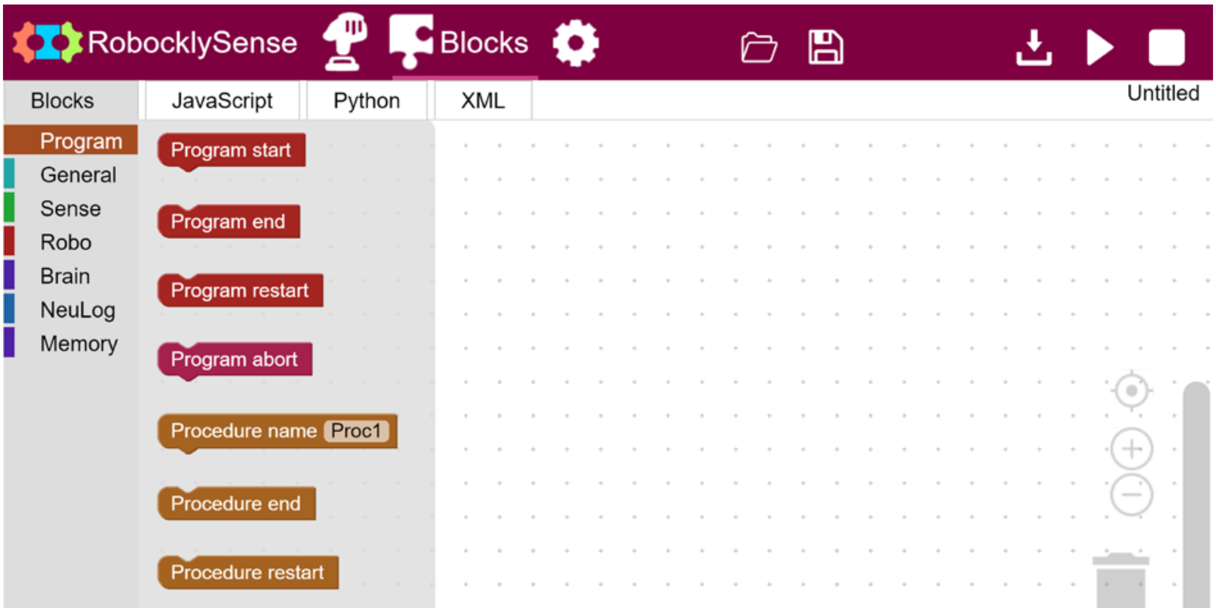
### 1.2.1 First program – forward

1. Move to **Block** mode.

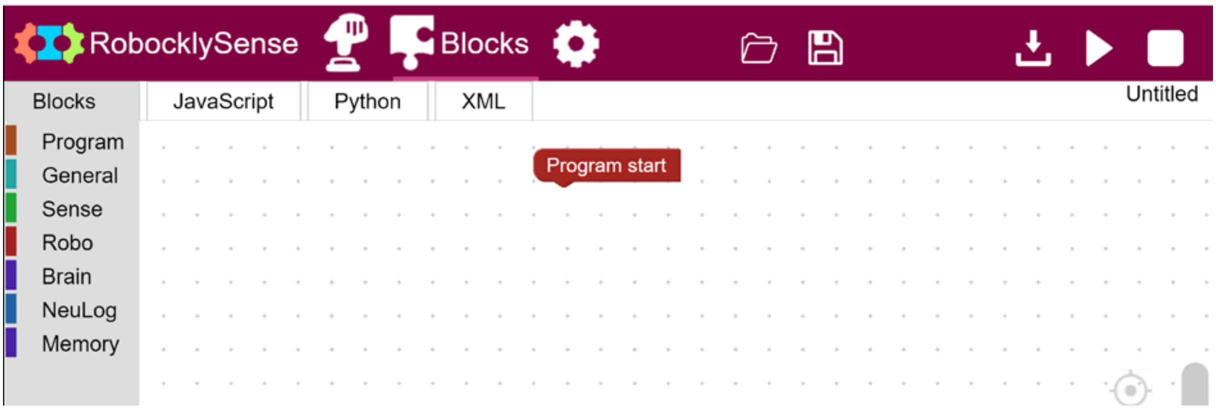



A computer program is composed of chains of instructions.

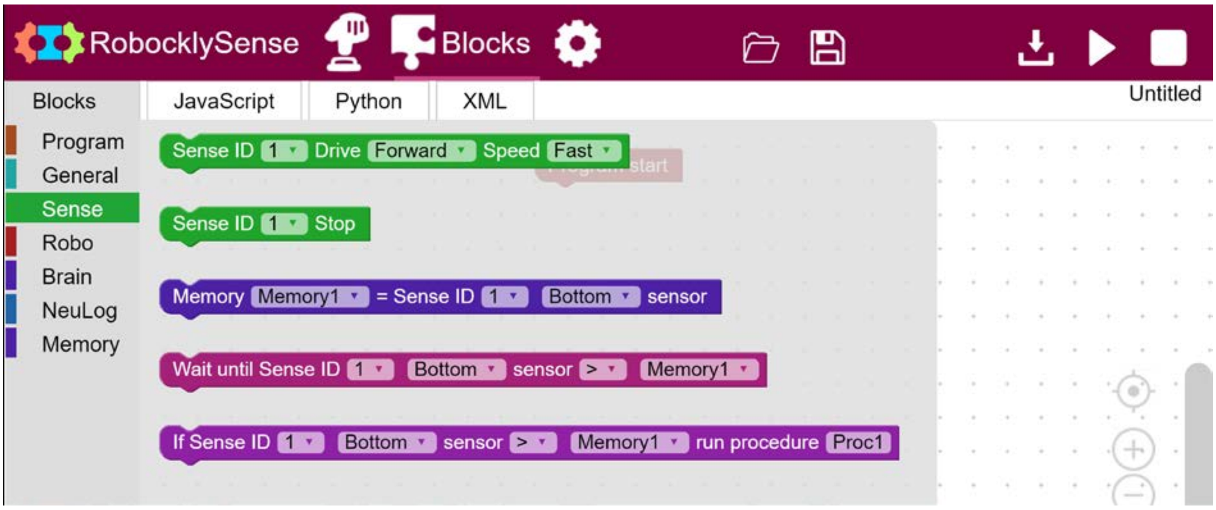
2. Click on the **Program** button and a list of program instruction list will appear:



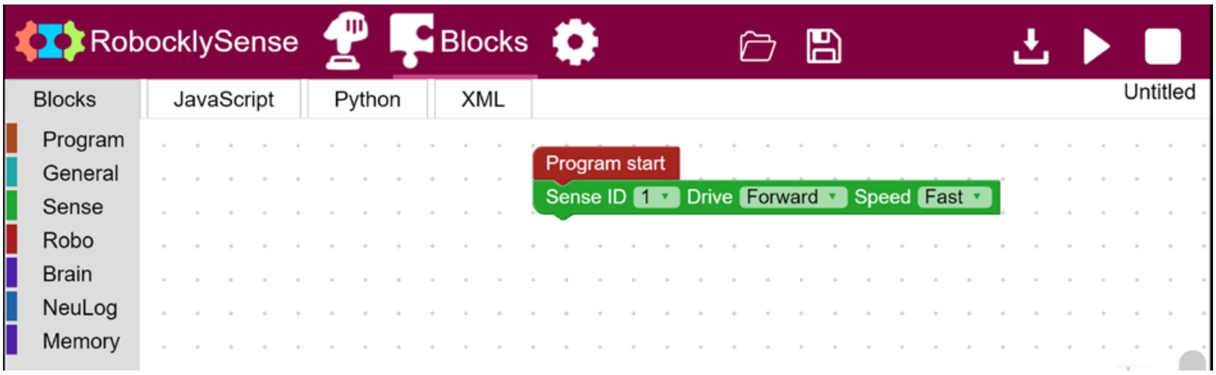
3. Click on the **Program start** instruction block and drag it to the right.




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




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



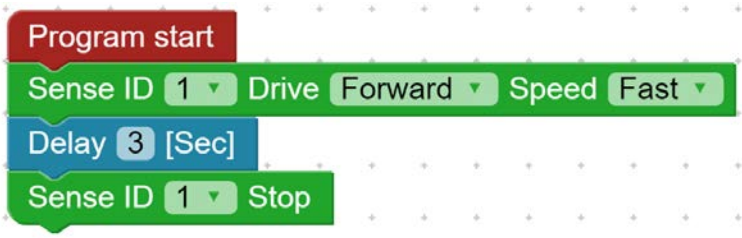
6. Click on the **General** button and select the **Delay** instruction block.

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


7. Change the delay value to 3 seconds.

8. Click on the **Sense** button and select the **Stop** instruction block.

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




   The **Stop** instruction will be executed 3 seconds after the **Drive Forward** instruction.

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

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

10. Check that you have the following program:


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

### 1.2.2 Program download

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

   We can download the program into its memory.

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

   This will transfer the PC program into the SENSE flash memory and will replace a previous saved program.

3. 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.


4. 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 button and you will see the SENSE move forward for 3 seconds and then stop.

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

   You may also have the NeuLog battery module BAT-202, which can be plugged directly into one of the SENSE sockets

   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:

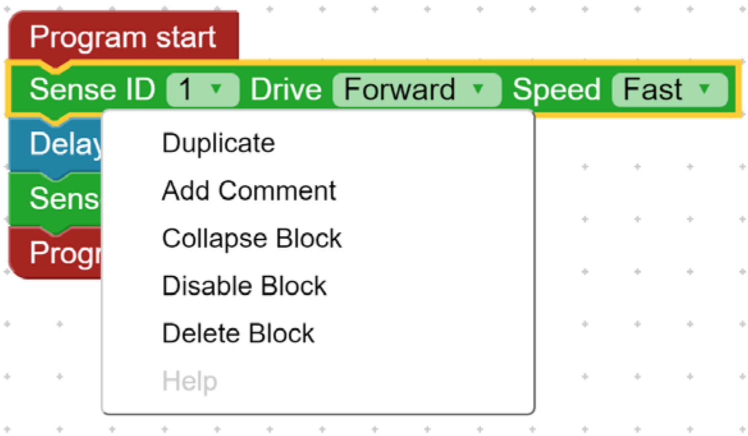
6. Press the **Start/Stop** button and you will see the robot move forward for 3 seconds and then stop.

   Please note: The SENSE LED blinks while running.

7. 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

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



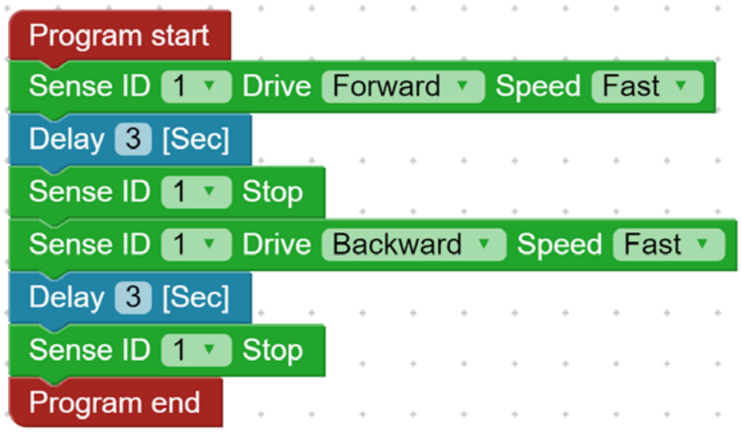

2. Select **Duplicate** and a new **Drive** instruction block appear. Attach it to the **Stop** instruction block.

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

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

4. Duplicate the **Delay** and the **Stop** instruction blocks.

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




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

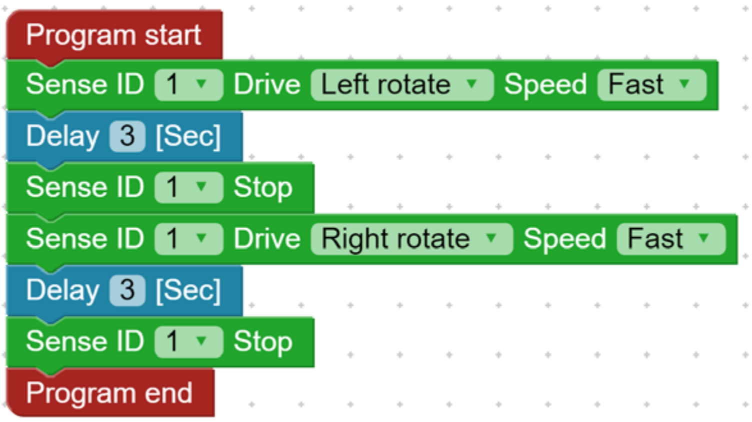
7. Click on the **Program Download** button.

8. 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

1. Change the program to the following one:




Pay attention to the **Drive** instructions.

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

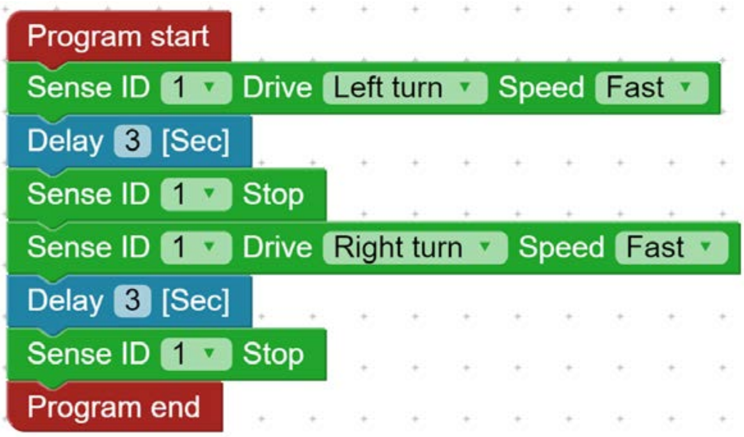
3. Click on the **Program Download**  button.

4. Click on the **Run**  button.

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

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

6. Change the program to the following one:




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

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

9. 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 behaviors of the two rotating programs?

10. Change the delay time for having turns of 90º.

## 1.2.5 Challenge exercises – Moving in a square

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 original 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 original 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 autonomous
- USB connection cable
- BAT-202 Battery module

## 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 example for closed loop system is a control system that lights up a 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 to use 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.

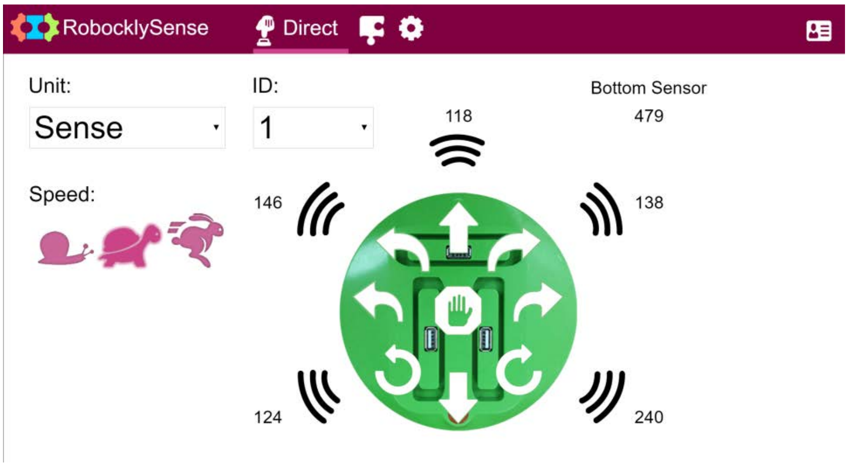
In **RobocklySense** we use memories (memory1 to memory9) as variables.

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.



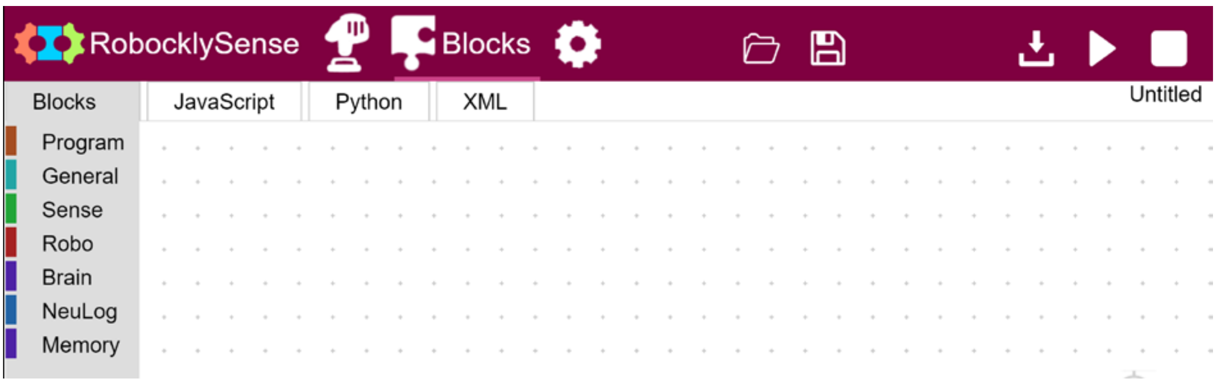

3. Test the SENSE motors as described in experiment 1.1.

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

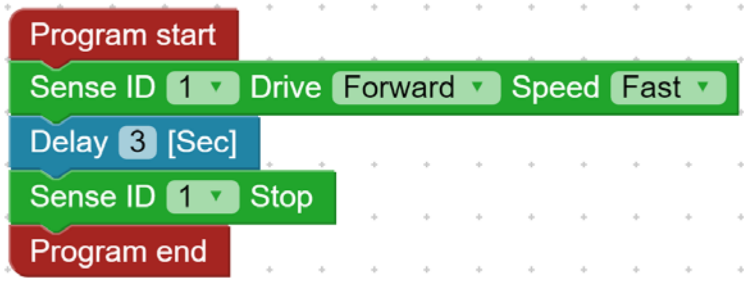
5. Record the values of the Bottom sensor when the SENSE is on a white surface. We shall call this value White Value.

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 stop.

12. Disconnect the SENSE from the PC and plug in the battery module (BAT-202).

13. Press the SENSE panel button and you will see the robot move forward for 3 seconds and then stop.

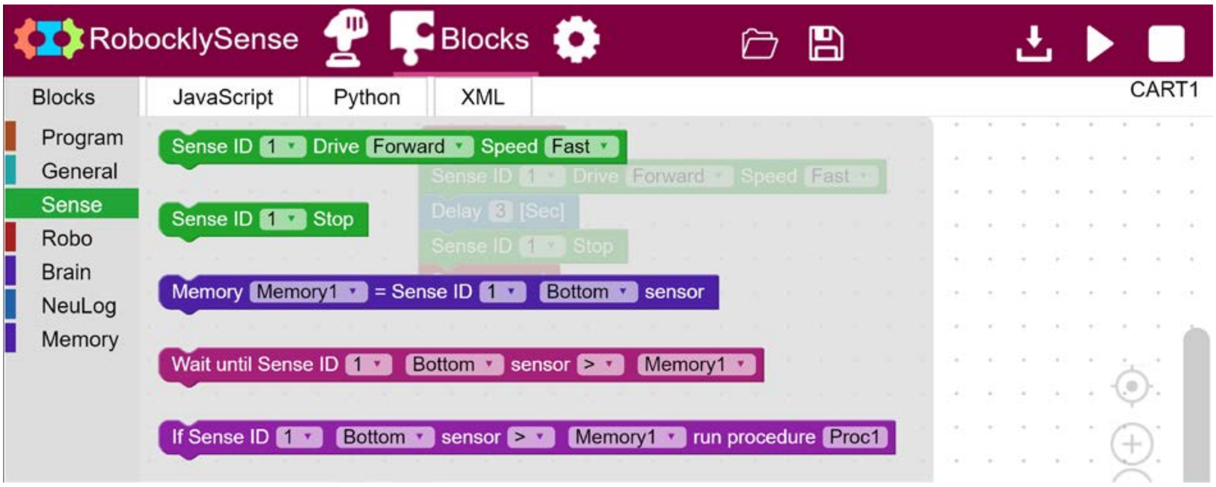
    The SENSE LED blinks while running.

14. Connect the SENSE to the PC.

### 1.3.2 Wait until

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

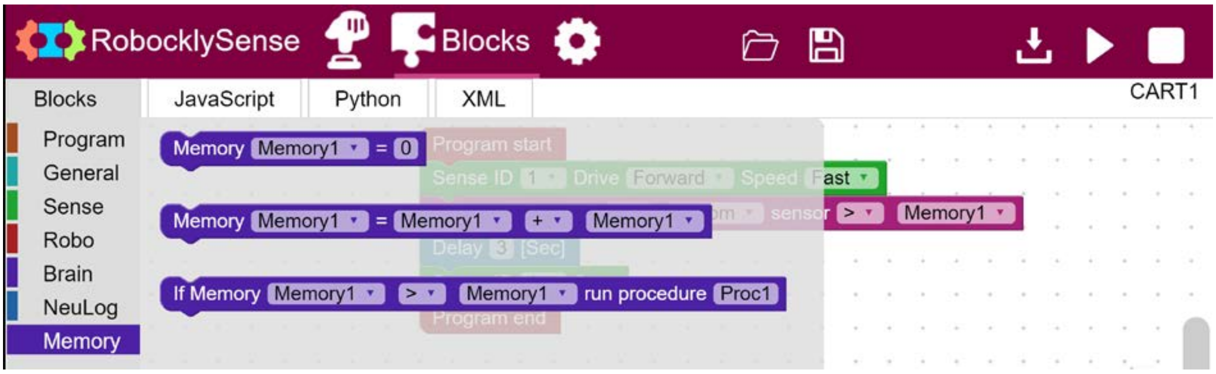
1. Click on the **Sense** button and a list of input instructions appear:




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

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

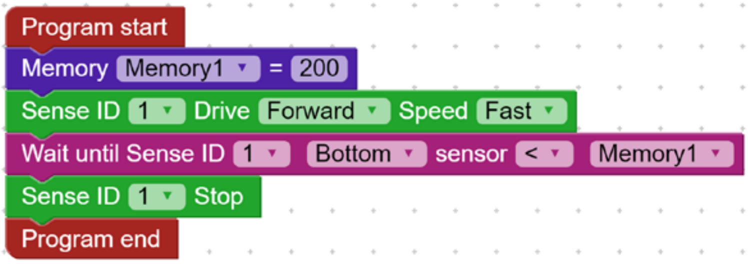
4. Click on the **Memories** button and a list of input instructions appear:




5. Click on the '**Memory [memory1] = 0**' instruction block and drag it to be the first instruction in the program (under the **Program start** block).

6. Change the number in the instruction to a value that is 50 above the **Black Value**.

7. Delete the **Delay** instruction block and check that you get the following:




8. Place the SENSE on a white surface with a black line on it.

9. Run the program.

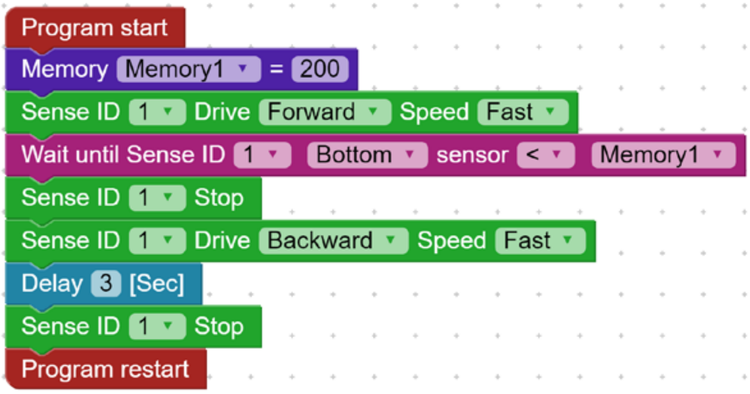
   The SENSE should move forward and stop when it reaches the black line.

## 1.3.3 Endless loop

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

1. 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.

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




3. Place the SENSE on a white surface with a black line on it.

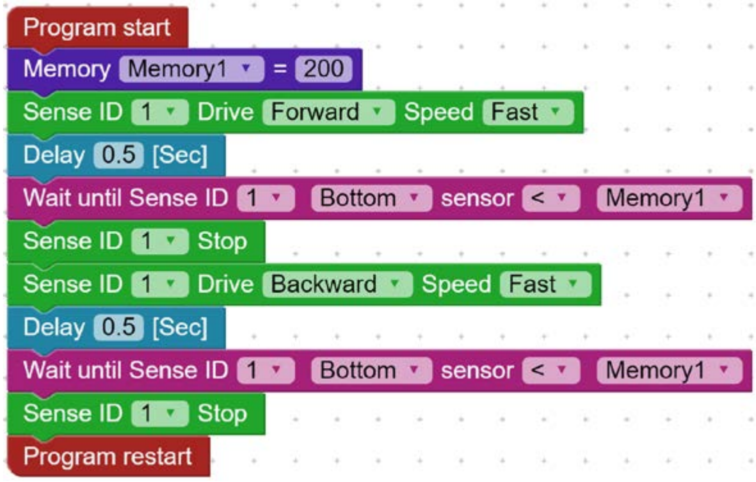
4. Run the program.

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

5. Stop the program by pressing the SENSE pushbutton or by clicking the **Stop** button, if the SENSE is connected to the computer.

### 1.3.4 Movement between two lines

1. 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.

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

3. 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 – Between a wall and a line (I)

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
- USB connection cable
- BAT-202 Battery module

## Discussion:

A computer program is 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. This way, when we run the program, the computer knows where to start.

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

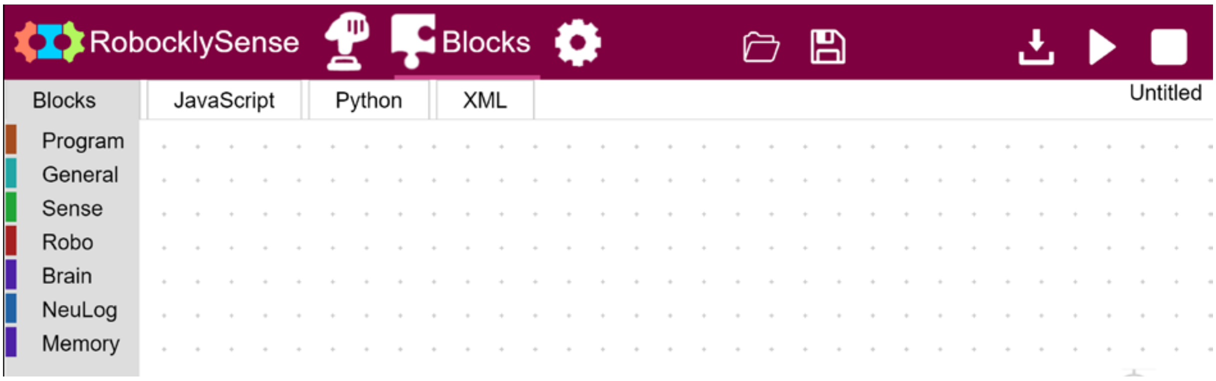
At the end of the program we need to add an instruction, which returns the program to the beginning (when we want the program to be executed repeatedly), or stops the program and returns to the operating system.

# Procedure:

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

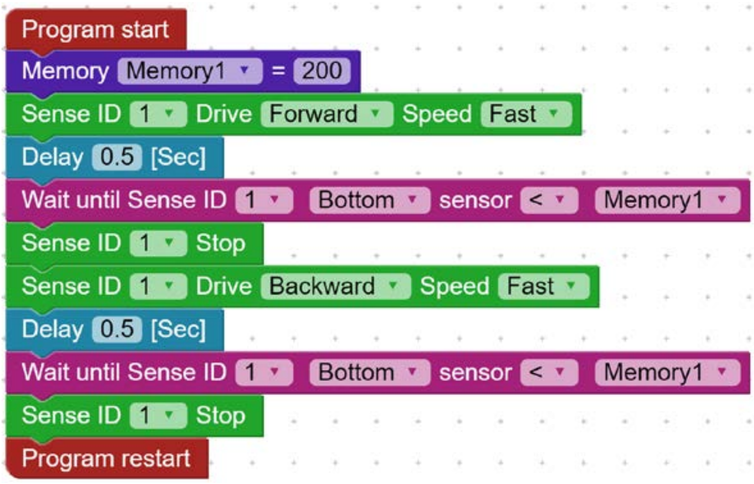
2. Run the **RobocklySense** software.

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. Drag the program to the **Trash box** in order to clear the screen.

## 1.4.1 Programs and procedures

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

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

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

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

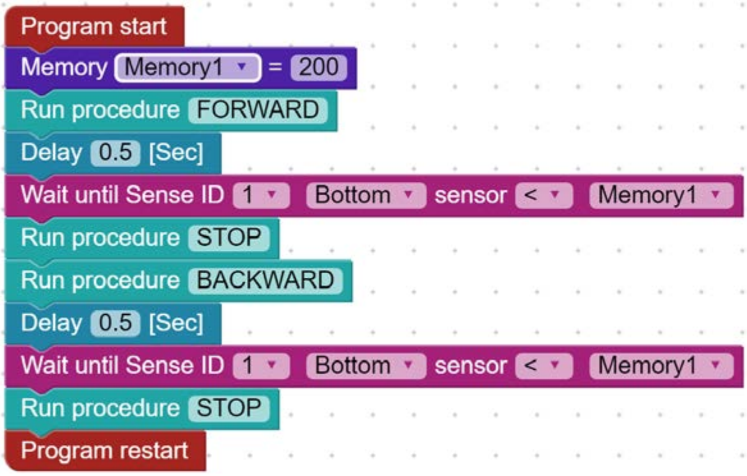

**<u>Note:</u>**
A procedure ends with **Procedure end** or **Procedure start**.

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


**<u>Note:</u>**
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.

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

7. 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 up more options as we shall see later.

8. 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.

   The program functions the same as the **CART2** program.

9. 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.

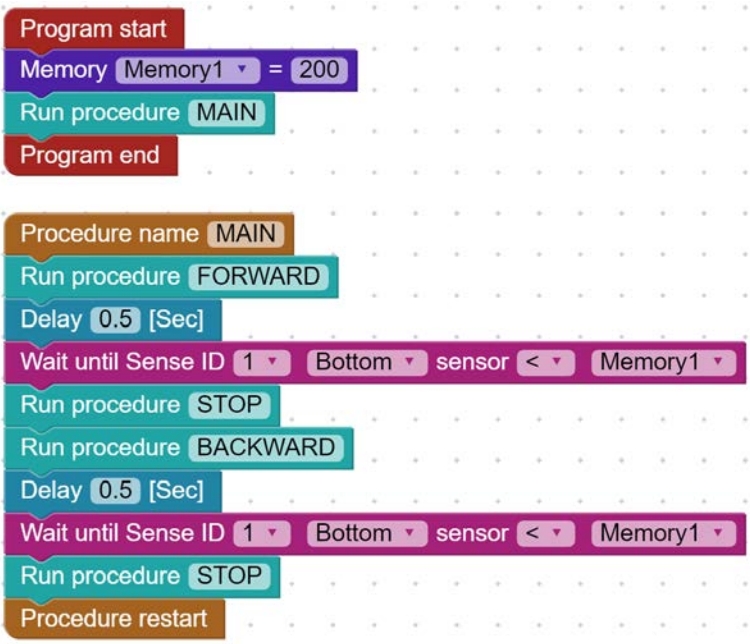
10. Stop the program.

## 1.4.2 Definitions

The definition of **memory1** is done in every loop of the program; we only have to do it once. It also slows 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 declared in the program (under the **Start program** block). After the declarations, the program runs the main procedure.

1. Change the program to the following program that includes a **MAIN** procedure:




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

3. The program functions the same as the **CART**3 program.

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

   The SENSE should run back and forth between the lines.

5. Stop the program.

## 1.4.3 Challenge exercises – Between a wall and a line (II)

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

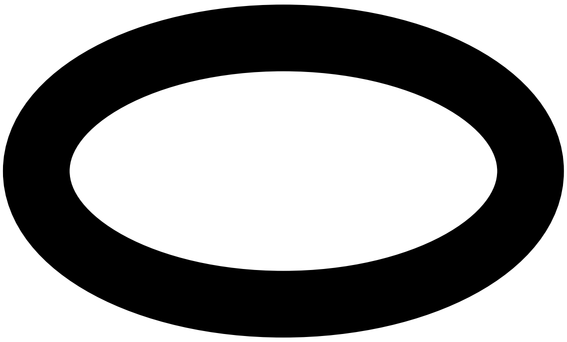
To move the SENSE along a black line we use slow turn procedures of the SENSE.

In slow turns, one wheel rotates 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.

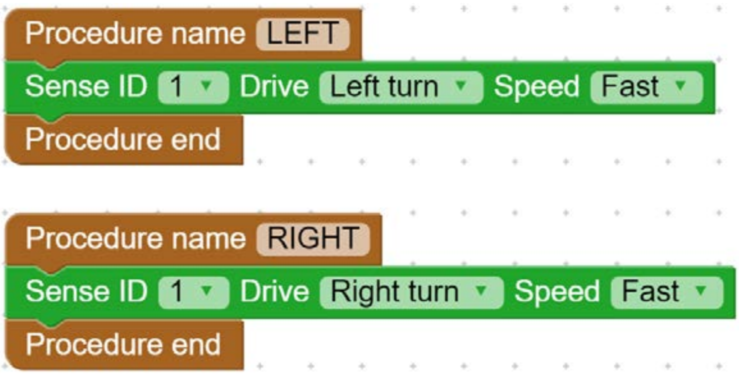
1. Print on a full page a black line as the following:



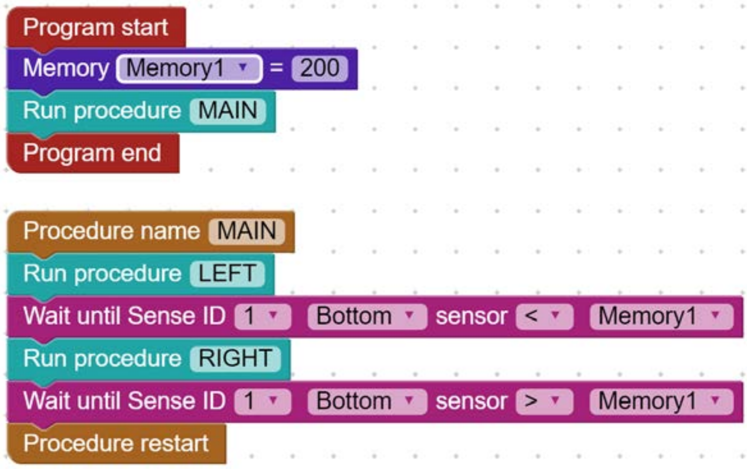
The width of the line should be at least 4cm.

2. Clear the screen.

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



4. Create the following main program which includes a **MAIN** procedure:




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

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

6. Put the SENSE near the black line.

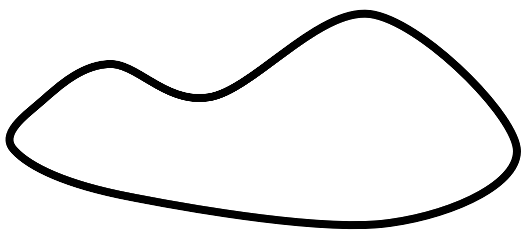
7. Run and check the SENSE movement.

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

## 1.4.5 Challenge exercise – Along a complex black line

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

Example of a complex line:



# Experiment 1.5 – Conditions and Decisions

## Objectives:

- The IF instruction.
- OR condition.
- AND condition.

## Equipment required:

- Computer
- RobocklySense software
- SENSE autonomous
- USB connection cable
- BAT-202 Battery module

## Discussion:

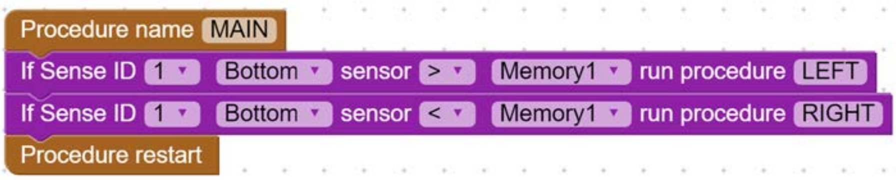
## 1.5.1 If – then instruction

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 a procedure to operate when the condition exists.

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

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



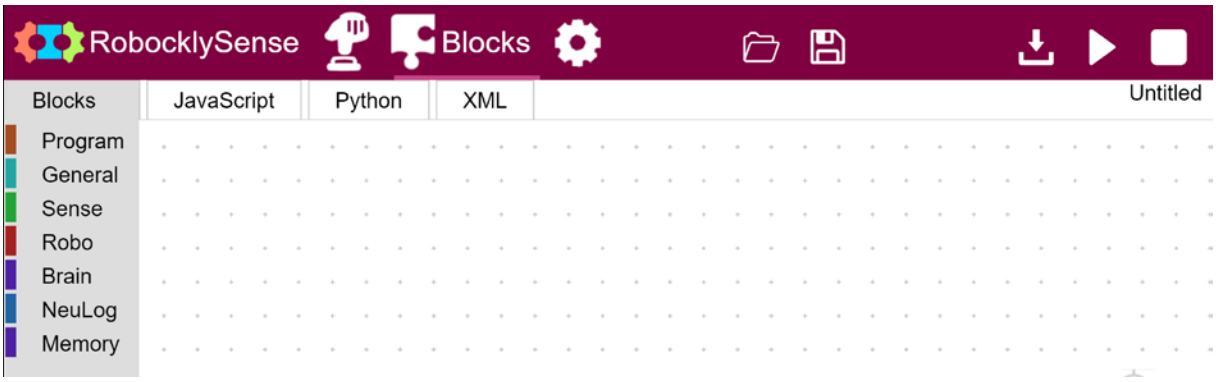

We can create complex conditions, called AND condition and OR condition, explained in the experiment later.

# Procedure:

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

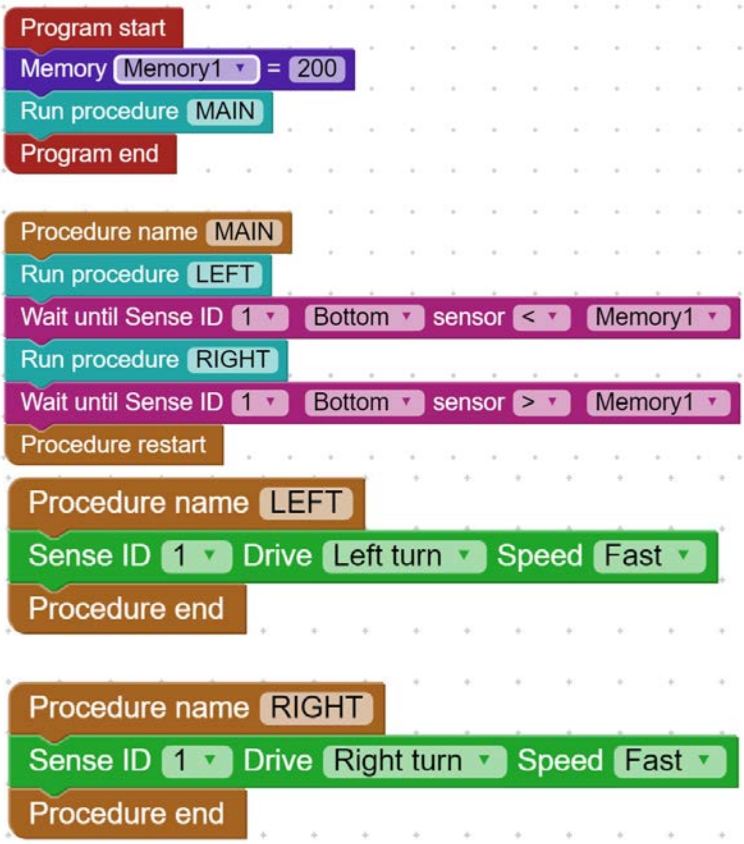
2. Run the **RobocklySense** software.

3. Move to **Block**  mode.



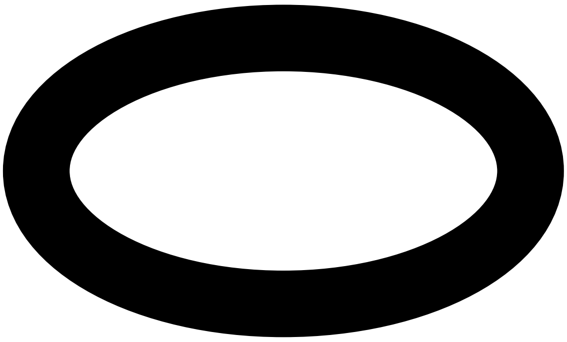

4. Click on the **Open**  button and open the program **CART**5.

5. Check that you have the following program:




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

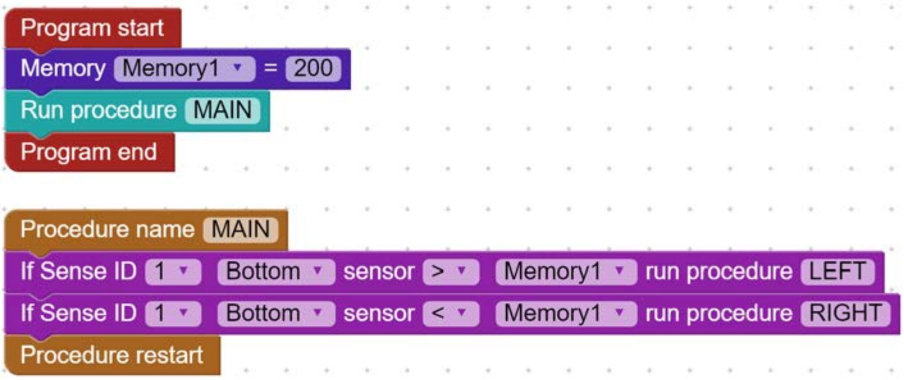
6. Place the SENSE on the black line circle and run the program.



The SENSE should go along the black line.

7. Stop the program.

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




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

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

10. Put the SENSE near the black line.

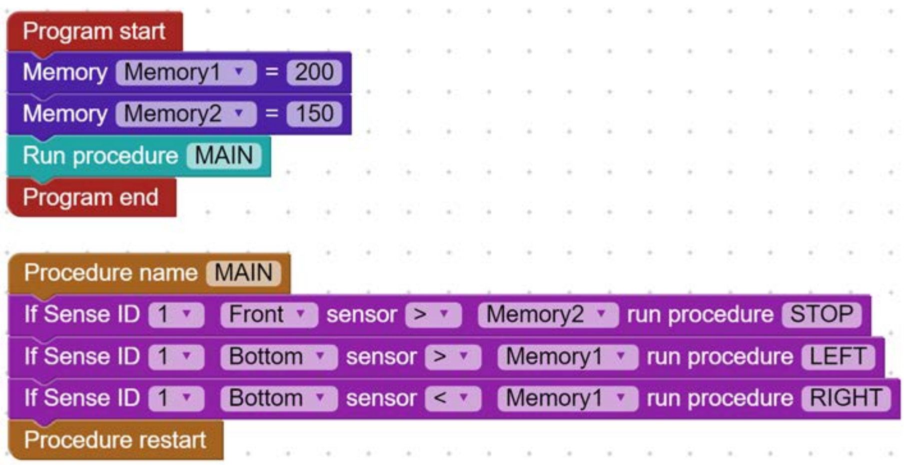
11. Run and check the SENSE movement.

12. 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.

13. Build the following **STOP** procedure:


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




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

In every cycle, the main procedure checks the distance from the wall and calls the **STOP** procedure when the SENSE is close to the wall.

15. Put the SENSE near the black line.

16. Run and check the SENSE movement.

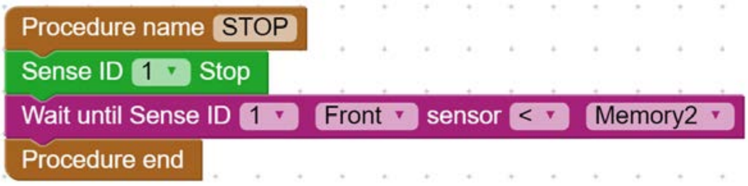
17. Put your hand in front of the SENSE, while it moves.

Instead of stopping, the SENSE just slows down.

Think why.

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

    Change the **STOP** procedure to the following:




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

20. Put the SENSE near the black line.

21. Run and check the SENSE movement.

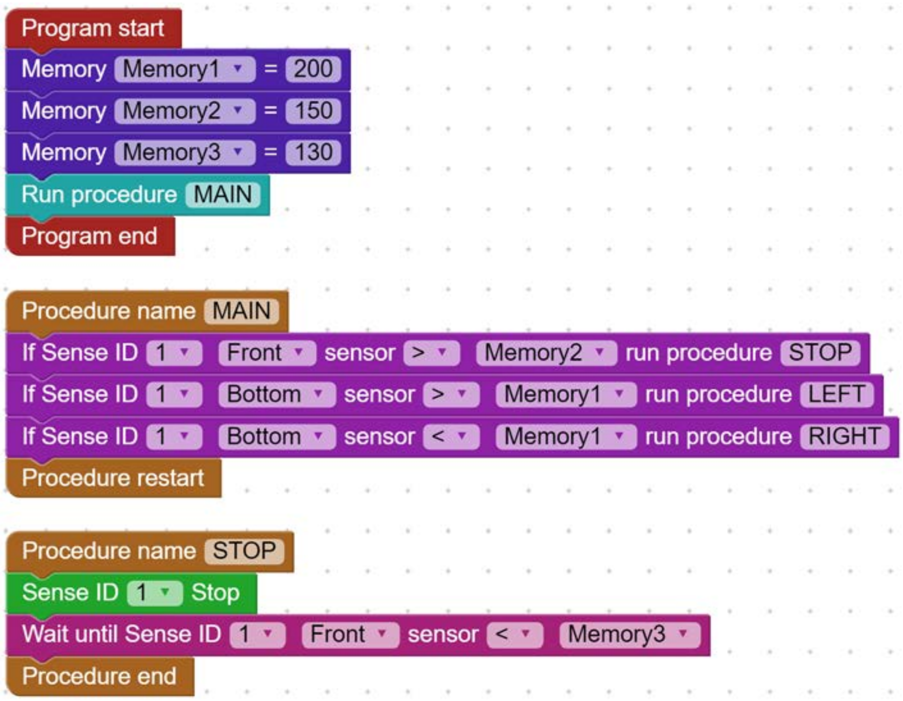
22. Put your hand in front of the SENSE, while it moves.

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

### 1.5.2 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. The reason for that is because we want to avoid having the system "bounce".

1. Change the program and procedures to the following:

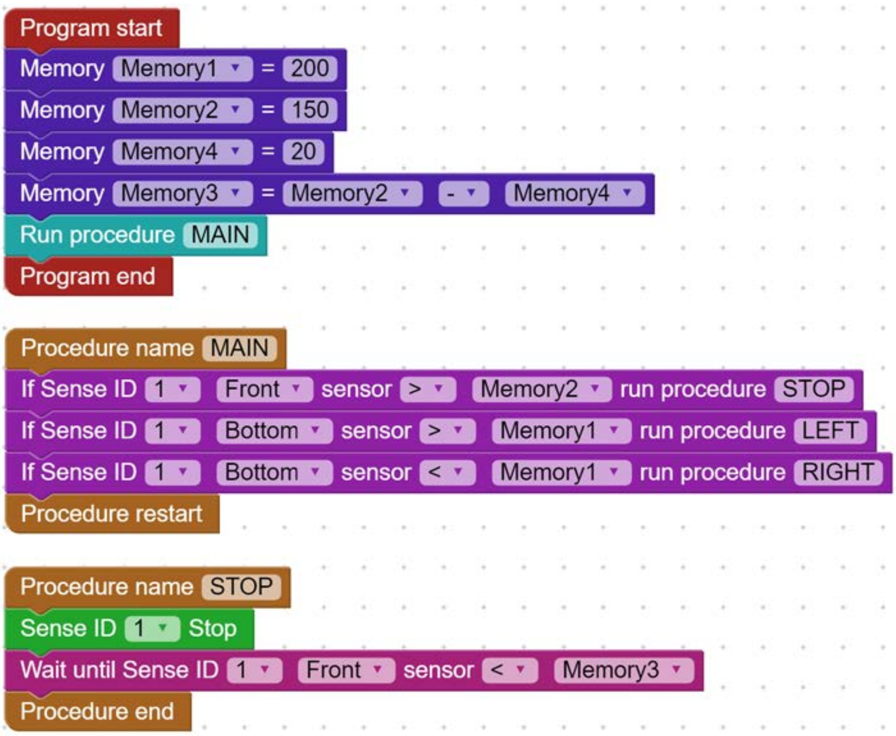



2. The **STOP** value is higher than the **WAIT** value.

3. Run and test this program.

4. When we change the **STOP** value, we have to change the **WAIT** value.

   The following program makes sure that the **WAIT** value (memory3) will always be lower than the **STOP** value (memory2).




5. Run and test this program.

### 1.5.3 AND condition

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

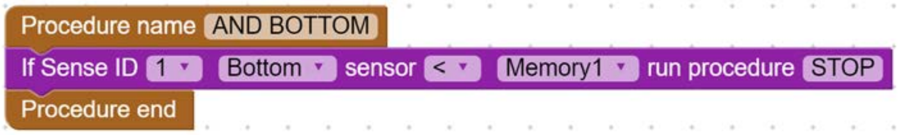
To achieve that we need the **AND** condition.

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

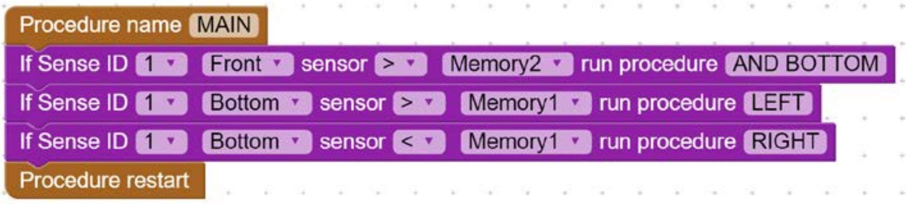
The RobocklySense does not have this option.

The AND operation can be achieved by creating another procedure.

1. Create the **AND BOTTOM** procedure with a single instruction:




2. Change the main procedure to the following one:




3. Analyze the program and the procedures.

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

5. Put the SENSE near the black line.

6. Run and check the SENSE movement.

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

   Check the SENSE behavior.

### 1.5.4 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 on a white surface **or** when it is close to the wall **or** both.

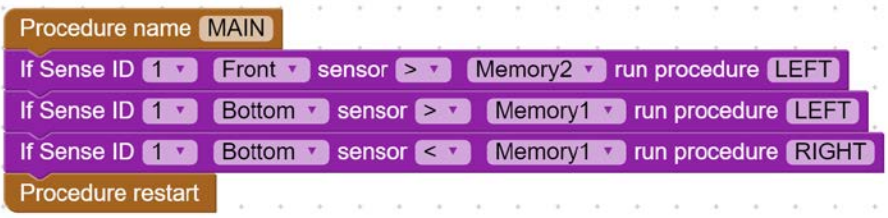
To achieve that we need the **OR** condition.

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.

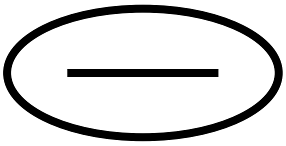
1. Change the main procedure to the following one:




2. Analyze the program and the procedures.
3. Save the program under the name **CART9**.
4. Put the SENSE near the black line.
5. Put a block on the black line.
6. Run and check the SENSE movement.

## 1.5.5 Challenge exercise – Along two lines

Task 1: Prepare two black lines as follows:



Put an 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 along it.

## 1.5.6 Movement along a wall

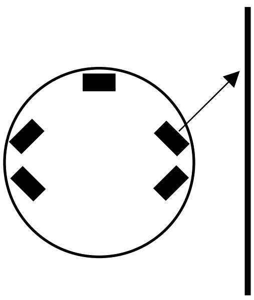
To move the SENSE along a wall, we use the same algorithm of moving the SENSE along a black line. We use the slow turn procedures.

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 $45^o$ to the SENSE base.

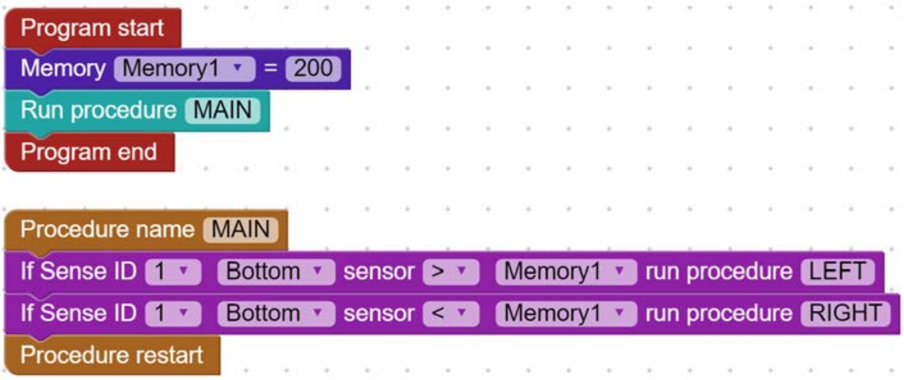


When the SENSE turns to the right, the measured distance is smaller than when it turns to the left.

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

1. Take a round box (cylinder) to be used as the first exercise wall.

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

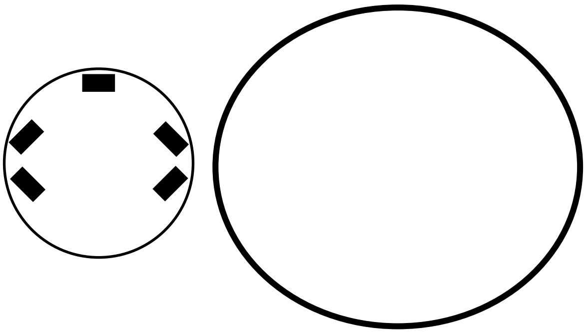



3. Change the main procedure of **CART6** to use the **Right Front sensor** instead of the **Bottom sensor**.

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

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

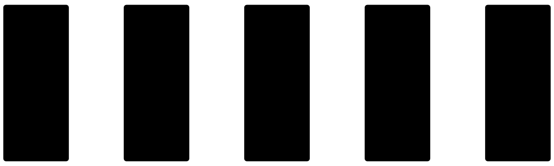
5. Put the SENSE near the round box and run the program.



   Check that the SENSE moves around the box.

# Challenge 1.6 – Counting

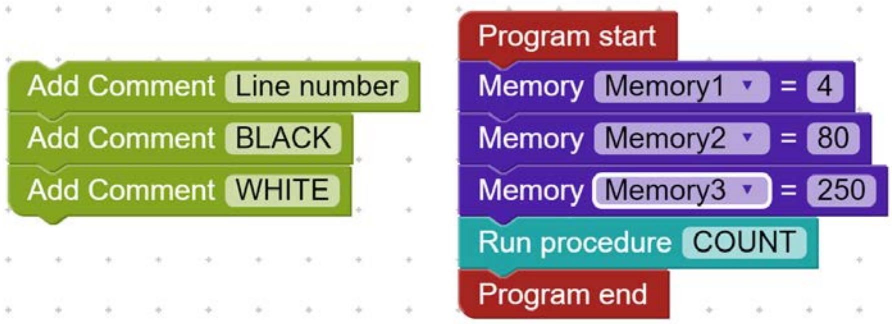
Draw block lines on a white paper.



Create a program that moves the robot through the block lines and make it stop on the fourth line.

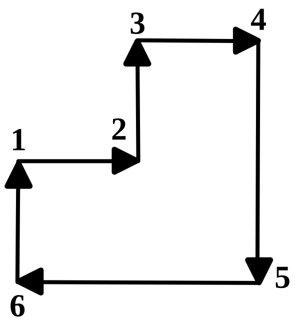
Use variables to count the lines.

The Program start should be as the following:



# Challenge 1.7 – Automatic movement

Create a program that moves the robot according to the following figure:

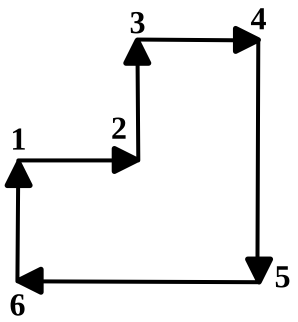


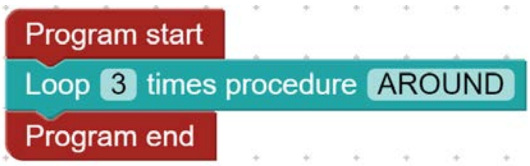

Create an AROUND procedures that moves the cart from point to point.

The Program start should be as the following:


# Challenge 1.8 – Loops

Use loop commands to make the robot do the following cycles 3 times.



# 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, units 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.

Every brain unit has an ID number. Every message from the control unit starts with ID number. Only the brain unit with this ID number interprets the message and executes it.

This system construction is the way modern systems are built, and has important advantages:

1. It creates a system with much less wires. The wires go from one module to another and not from all modules to the control unit.

2. This kind of system can easily be changed and expanded, and does not depend on the control units number of inputs and outputs.

The experiments in this chapter use the following brain units:

- Neulog light sensor (NUL-204)
- Neulog sound sensor (NUL-212)
- Neulog motion sensor (NUL-213)
- Neulog magnetic sensor (NUL-214)
- Brain tracking unit (SNS-101)
- Brain IR transmitter (SNS-160)
- Brain gripper arm (SNS-167)

# 2.2 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.

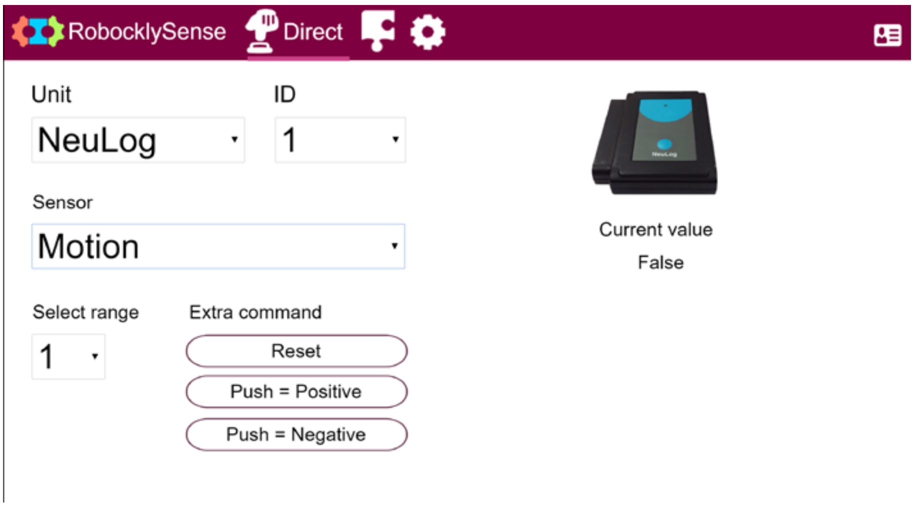
The SENSE has three sockets for NeuLog sensors.

# 2.3 Changing Brain unit ID

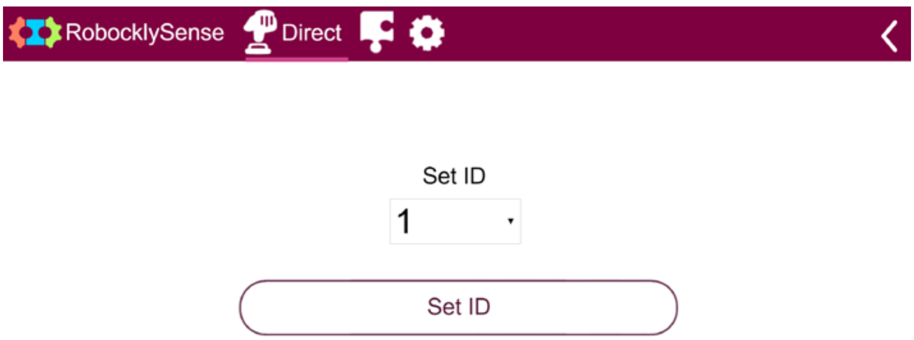
As said before, every brain unit has ID number. The ID number enables us to use up to nine brain units of the same kind. We just have to take care that each one of them will have different ID number.

In **Direct** mode screen, we have a special icon for changing the ID number of the unit.

The following screen is the **Direct** screen of the motion sensor.




Clicking on the  icon on the right, will show the following screen:




In order to change the ID number of the unit, we have to connect only one unit to the PC, to set the required ID number in the Set ID field and to click on the **Set ID** button.

# Experiment 2.1 – Sound Sensor

## Objectives:

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

## Equipment required:

- Computer
- RobocklySense software
- SENSE autonomous
- USB connection cable
- BAT-202 Battery module
- NUL-212 NeuLog sound sensor

## Discussion:

The sound sensor uses an internal microphone and special amplifier. 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 indicates a number according to signal shape. 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.

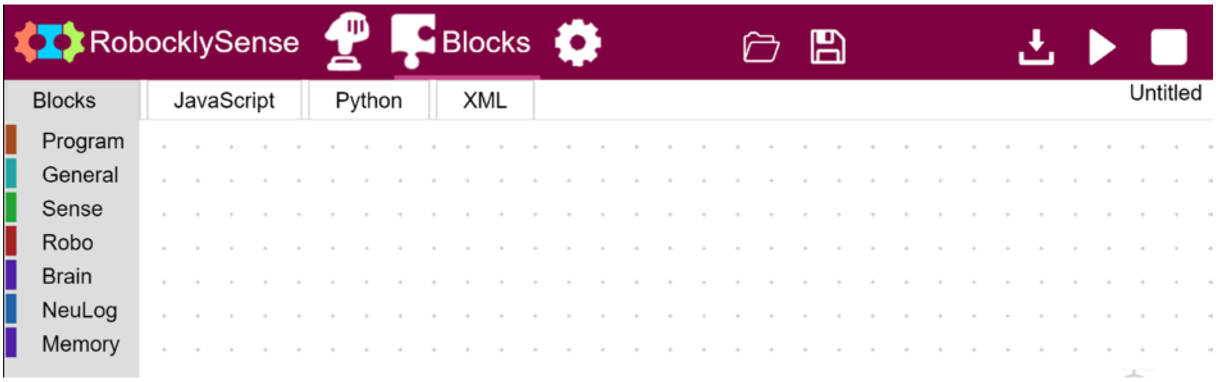
# Procedure:

1. Plug the NUL-212 sound sensor into one of the SENSE sockets.

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

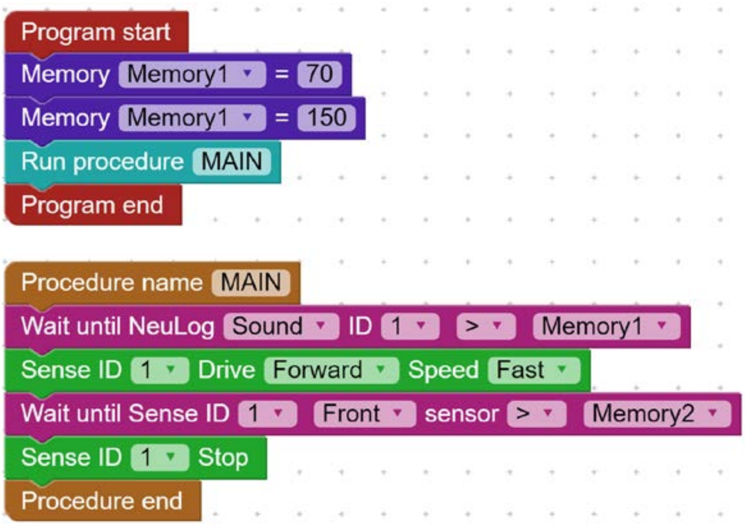
3. Run the **RobocklySense** software.

4. Move to **Block** mode.




Create a program that waits for a sound level of 70 dB, moves the SENSE to a wall, and then stops.

5. Build the following program and its main procedure:




6. Observe the program and make sure that you understand all of 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 towards the wall and stop.

### 2.1.1 Challenge exercise – Wait for sound

Task 1: Improve the program so:

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

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

(c) Returns 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
- USB connection cable
- BAT-202 Battery module
- NUL-213 NeuLog motion sensor

## Discussion:

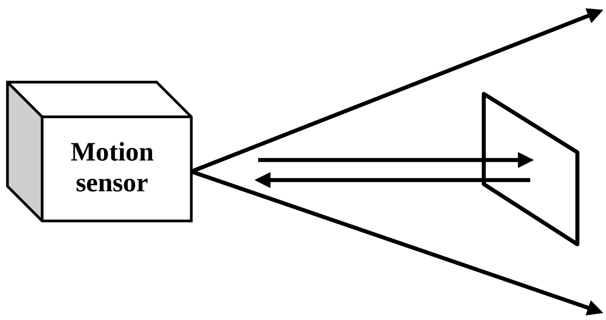
The motion sensor uses an ultrasonic transducer to both transmit an ultrasonic wave, and to measure its echo return. Objects in the range of 0.15 to 10 meters can accurately be measured to give distance, velocity, and acceleration readings using this method.

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 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. These pulses 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 calculated by the difference between two successive distances divided by the time between the samples (according to the sampling rate).

Acceleration is calculated 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 with the NeuLog software.

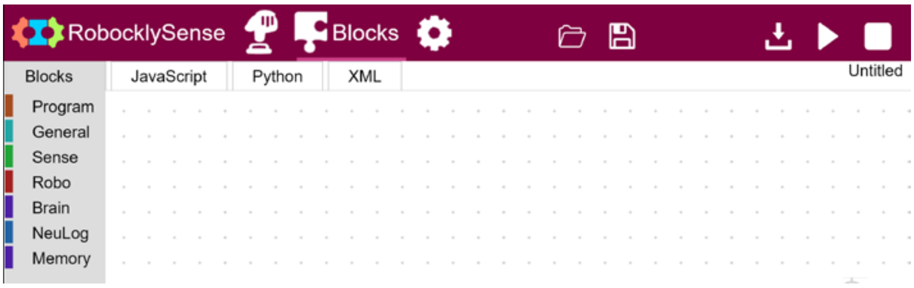
# Procedure:

1. Plug the NUL-213 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.

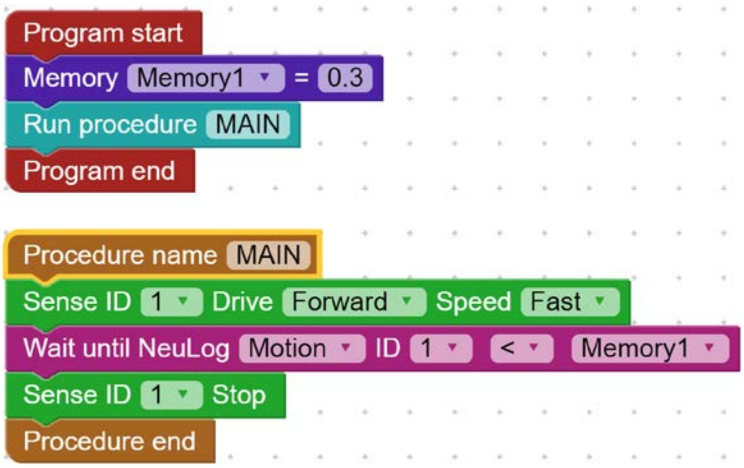
3. Run the **RobocklySense** software.

4. Move to **Block**  mode.




Create a program that moves the SENSE forward to a wall and stops 30 cm away from it.

5. Build the following program and its main procedure:




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

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

The SENSE should move to the wall and stop 30cm away from it.

## 2.2.1 Challenge exercise – Moving in a distance range

Description: Going forward towards a wall, stop 30cm before the wall, then go backward and stop at 50cm from the wall and return.

Task 1: Improve the program so the SENSE will:

- move towards the wall,
- stop 30 cm in front of it,
- wait for 2 seconds,
- go backwards until a distance of 60 cm,
- stop for 2 second,
- return to the beginning.

# Experiment 2.3 – Brain Tracking Unit

## Objectives:

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

## Equipment required:

- Computer
- RobocklySense software
- SENSE autonomous
- USB connection cable
- BAT-202 Battery module
- SNS-101 Brain tracking unit
- SNS-160 IR transmitter

## Discussion:

### 2.3.1 IR Transmitter

The infrared 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 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 two IR (infrared) sensors that enables it to track the IR transmitter.

The two IR sensors are at the same line with an 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 binary number that describes the detection status of an IR transmitter. This number is converted to detection results as the following:

None (00) – No IR transmitter light<br>
Right (01) – IR transmitter light on the right<br>
Left (10) – IR transmitter light on the left<br>
Front (11) – IR transmitter light at front

# Procedure:

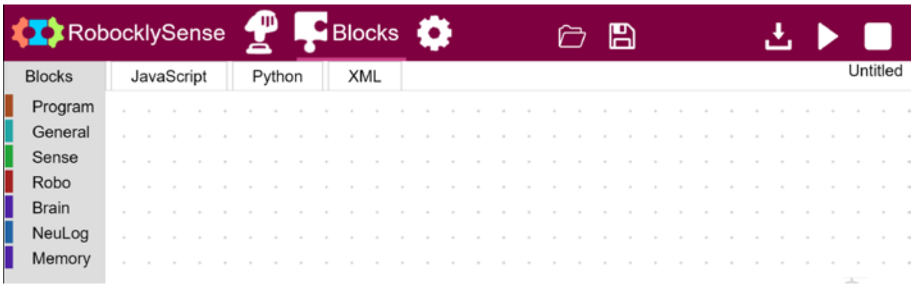
1. Plug the brain tracking unit into the front socket of the SENSE.

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.

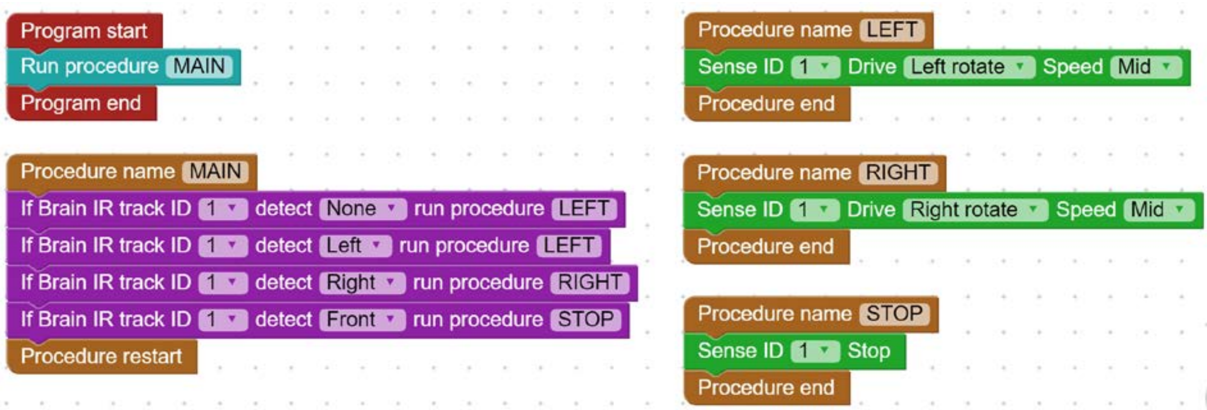
4. Run the **RobocklySense** software.

5. Move to **Block** mode.




Create a program that rotates 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 its main procedure:




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

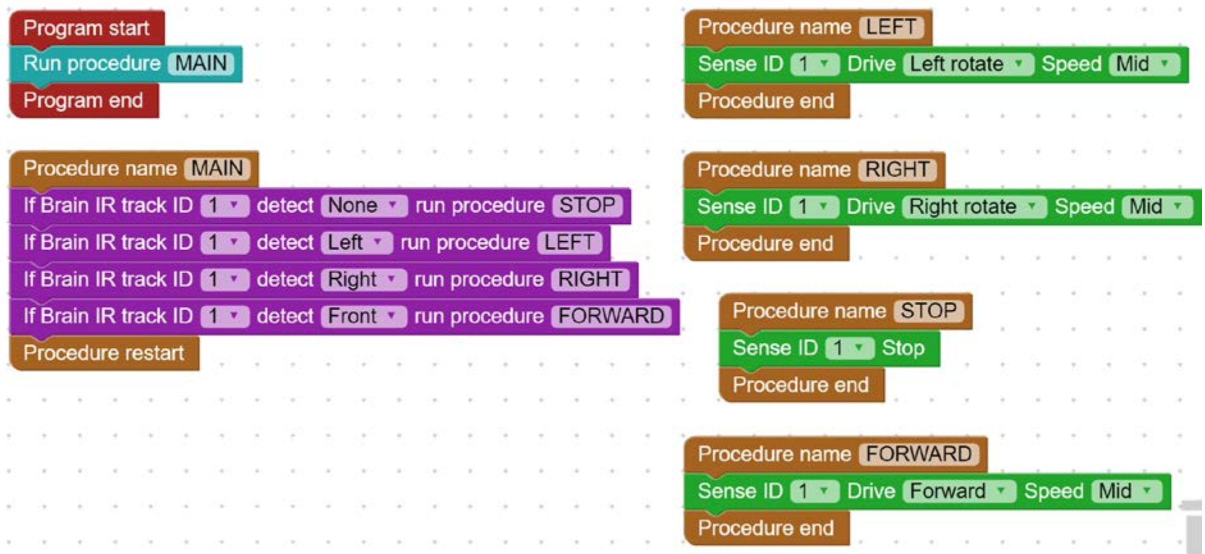
8. Download the program.

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. Plug the IR transmitter into battery module or power bank, move it slowly and check that the SENSE tracks it.

11. Change the program to the following:




12. This program should move the SENSE towards the IR transmitter. The SENSE waits when it does not detect the IR light. Check that.

## 2.3.3 Challenge exercise – Tracking the SENSE with an IR transmitter

Task 1: Improve the above program and procedures so the SENSE will stop in front of the IR transmitter.

Put the IR transmitter on a box or another SENSE that can be detected by the front sensor.

# Experiment 2.4 – Brain Gripper Arm

## Objectives:

- The brain gripper arm.
- Moving an object from one place to another.
- Drawing pictures with the brain gripper arm.

## Equipment required:

- Computer
- RobocklySense software
- SENSE autonomous
- USB connection cable
- SNS-167 Brain gripper arm
- A wooden rod
- A marker

## Discussion:

### 2.4.1 Brain gripper arm

The brain gripper arm has two servo motors. One servomotor moves the gripper up and down. The second servomotor opens and closes the gripper.

A servo motor is a motor with feedback. The feedback can be voltage according to the motor speed or the shaft angle, electrical pulses according to the motor shaft rotation and direction, and more.

Each servo motor of the gripper arm has transmission gear and potentiometer. The potentiometer consists of a variable resistor that creates variable voltage according to the servomotor shaft angle.

The brain controller of the gripper arm gets the required angle of the shaft. It turns the motor CW (Clock Wise) or CCW (Counter Clock Wise) until the potentiometer voltage suits this angle.

It checks the shaft angle all the time. If it changes mechanically, the controller will turn the motor ON to return the shaft to the right position.

# Procedure:

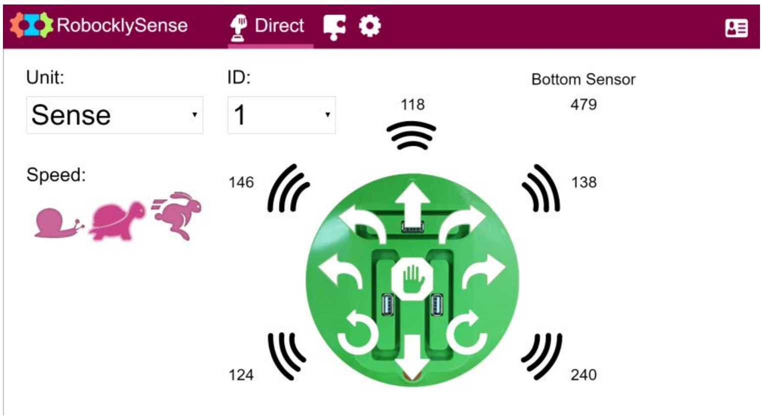
1. Plug the brain gripper arm into the front socket of the SENSE.

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

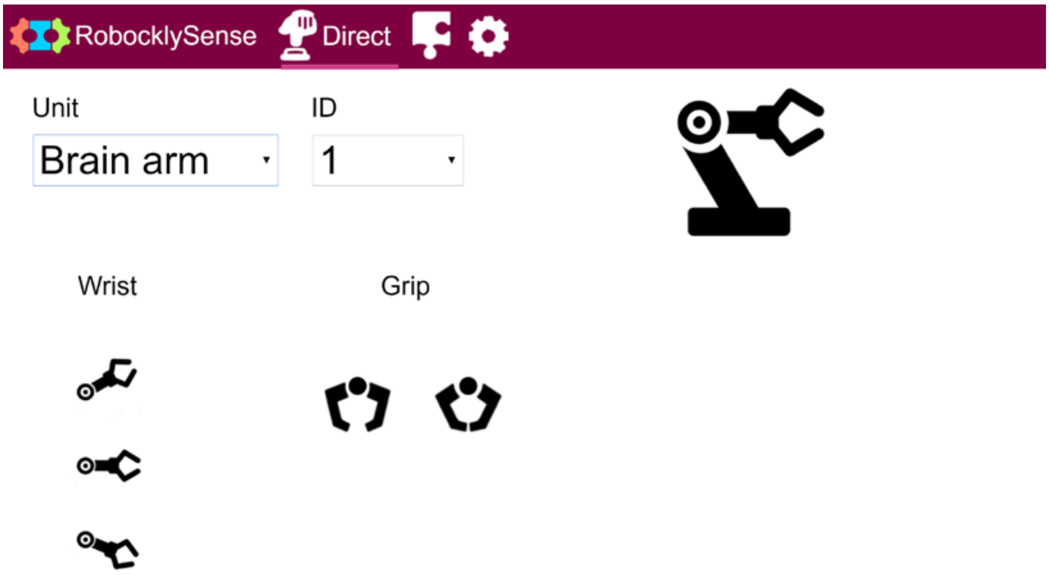
3. Plug battery to one socket of the SENSE.

4. Run the **RobocklySense** software.

   The **Direct** screen appears:




5. Change the selected Unit to **Brain arm**.




6. Play with the buttons on the screen.

   Move the gripper to up, mid and down positions.

   Open and close the gripper.

7. Move the gripper to mid position.

   Open the gripper.

   Put the wooden rod between the gripper fingers.

   Close the gripper.

   Raise the gripper.

   The close function closes the gripper for two seconds, measures the angle of the gripper servo motor and holds it in this angle.

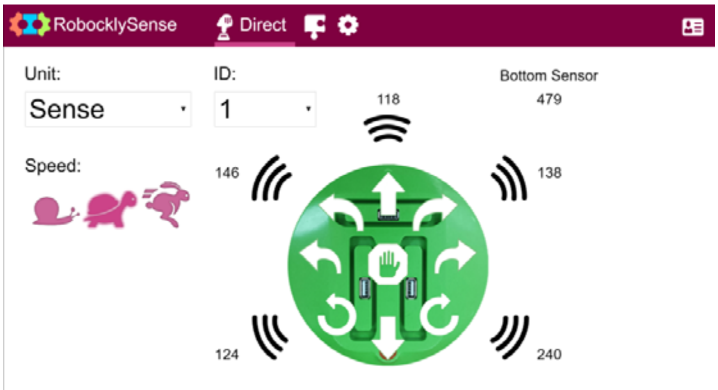
   This is why the gripper can hold objects with different thickness.

8. Move the gripper to mid position.

   Open the gripper.

   Put the wooden rod between the gripper fingers.

9. Change the selected Unit to **Sense**.

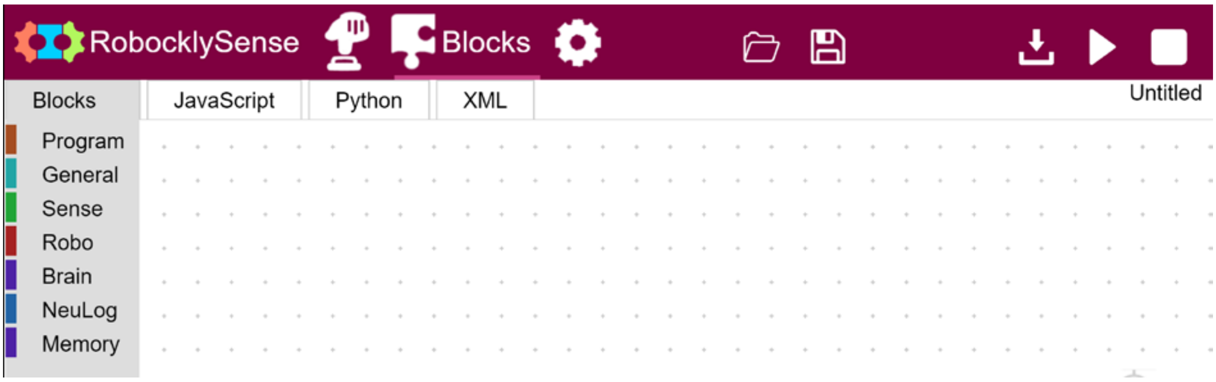



10. Record the value of the front sensor.

    For the following example, we shall use the number 200.

11. Put the wooden rod 20 cm in front of the SENSE.

12. Move to **Block**  mode.




13. Create a program that does the following:

    a. Opens the gripper
    b. Raises the arm
    c. Moves the Sense forward and stops when the wooden rod between the gripper fingers
    d. Lowers the arm to mid position
    e. Closes the gripper
    f. Raises the arm
    g. Moves forward for 2 seconds
    h. Lowers the arm to mid position
    i. Opens the gripper
    j. Moves backward for 2 seconds

14. Build the following program and its main procedure:


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

16. Download the program.

17. Disconnect the SENSE from the computer.

18. Press the SENSE **Run/Stop** button.

19. Check that the SENSE does its mission.

20. Change the program to run in endless loop.

21. Download the program, run and check the SENSE behavior.

## 2.4.2 Challenge exercises – the SENSE with gripper arm

Task 1: Change the last program to make the SENSE to rotate in about $90^o$ with the raised wooden rod before moving forward with it.

Task 2: Plug Sound sensor to the SENSE and make it wait for hand clapping before picking up the wooden rod.

Task 3: Make the gripper hold a marker manually.

Place the SENSE on wide white paper attached to the ground or to the desk.

Build some drawing programs.

# Experiment 2.5 – Robot and Science experiment

## Objectives:

- The Neulog light sensor.
- Running an experiment while moving.
- Using the SENSE as a USB module with Neulog software.

## Equipment required:

- Computer
- RobocklySense software
- SENSE autonomous
- USB connection cable
- BAT-202 Battery module
- NUL-204 NeuLog light sensor
- NUL-213 NeuLog motion sensor
- A flashlight

## Discussion:

### 2.5.1 The light sensor

The NUL-204 can be used for any science experiment where light intensity measurements are required, such as Chemistry, Physics, Biology, Environmental Science, etc.

This sensor can be used to take light measurements in low, medium and high light intensity environments, such as in classrooms and in open sunlight. The sensor can be used to measure both fast light changes like those produced by light bulbs connected to an AC supply, as well as the light intensity of a bulb or near steady levels outside on a sunny day.

The measurement unit for all three data collection ranges (low, medium, high) is the lux.

Lux (lx, or lux): The SI unit of light intensity.

NUL-204 includes a photodiode, which reacts with photons to release free electrons (photoelectrons). The amount of light striking the sensor is directly proportional to the voltage generated by the photoelectrons released. The sensor measures the general voltage released and thus calculates the light intensity.

If the light readout is very low, try changing the sensor's mode to a higher sensitivity. This is done by selecting the “Module setup” button on the light sensor module box in the NeuLog application.

The Neulog light sensor is able to adjust to 3 different sensitivity settings for ambient light because of its ability to change the internal hardware amplifier gain through the application.

Changing from illumination mode into signal mode is done automatically by the firmware according to the sampling rate.

### 2.5.2 Light intensity vs distance

In this experiment, we shall move the light sensor against a flashlight and a wall. We shall measure the light intensity and the distance with a distance sensor.

The **RobocklySense** has instructions to run an experiment and to stop it.

The NeuLog sensors are logger sensors. They sample and save the measurements in their flash memory.

# Procedure:

1. Plug the NUL-204 light sensor and the NUL-213 motion sensor units into the socket of the SENSE, as in the following picture.

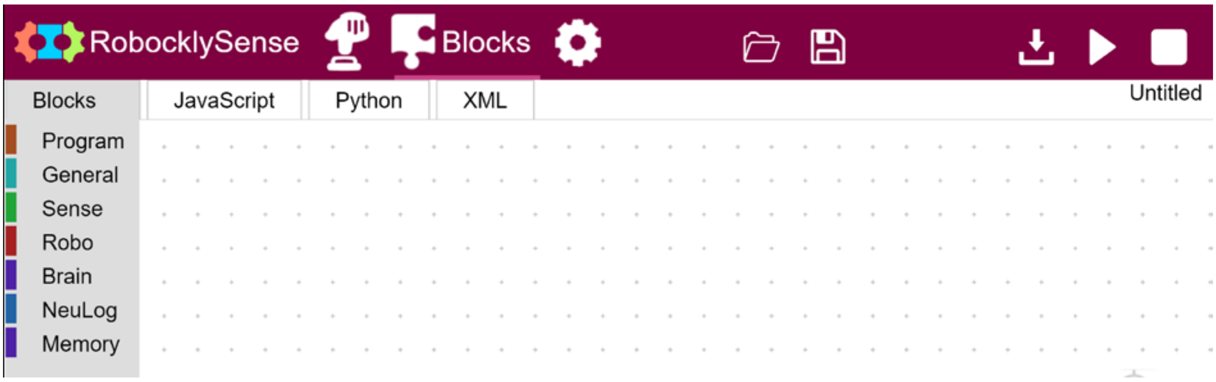
2. Plug the BAT-202 (Battery module) to the second socket.

3. Put a flashlight near the wall as in the following picture.

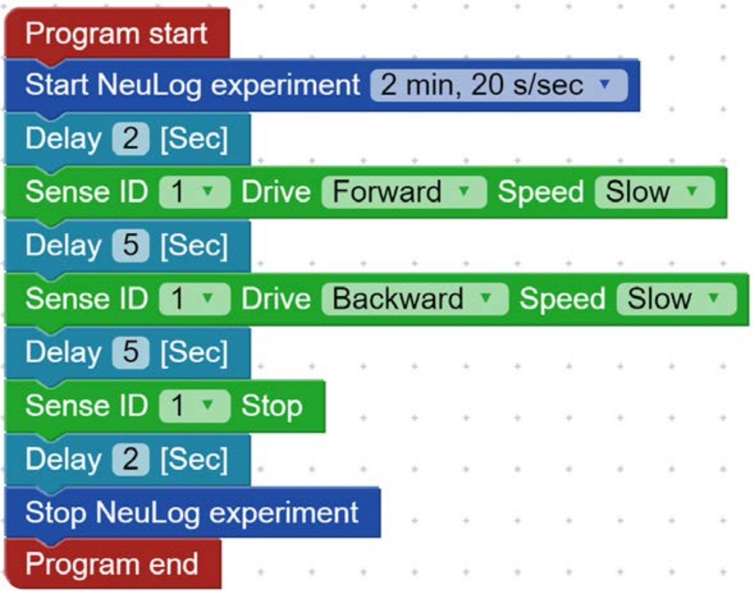
4. Connect the SENSE to the PC using the USB cable.

5. Run the **RobocklySense** software.

6. Move to **Block** mode.



7. Build the following program:




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

9. Download the program.

10. Disconnect the SENSE from the PC and place it on the floor against the flashlight as in the following picture.

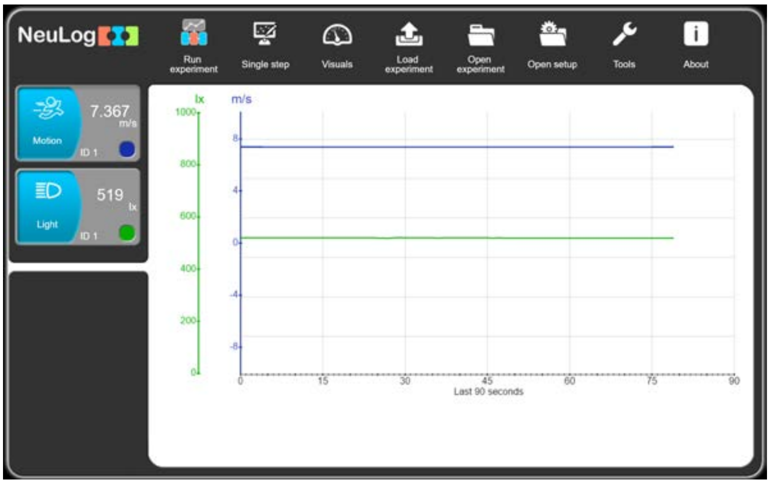
11. Run the program.

    The SENSE should wait for 2 seconds, go forward for 5 seconds, then go backward for 5 seconds and stop.

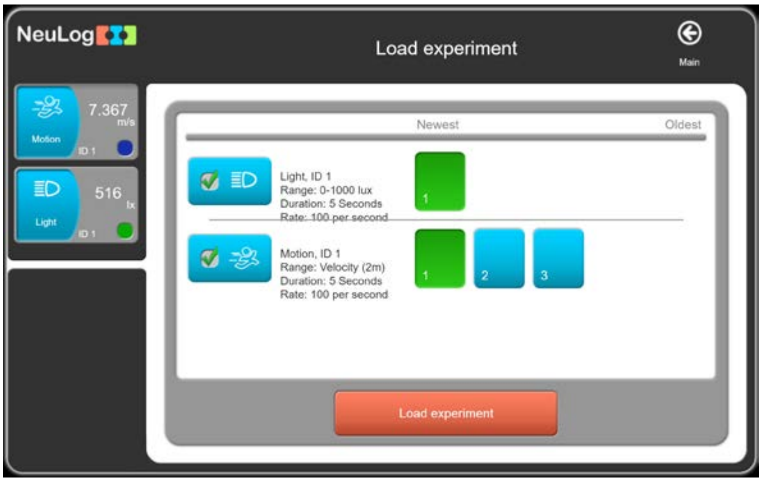
12. Connect the SENSE to the PC.

13. Exit the RobocklySense program.

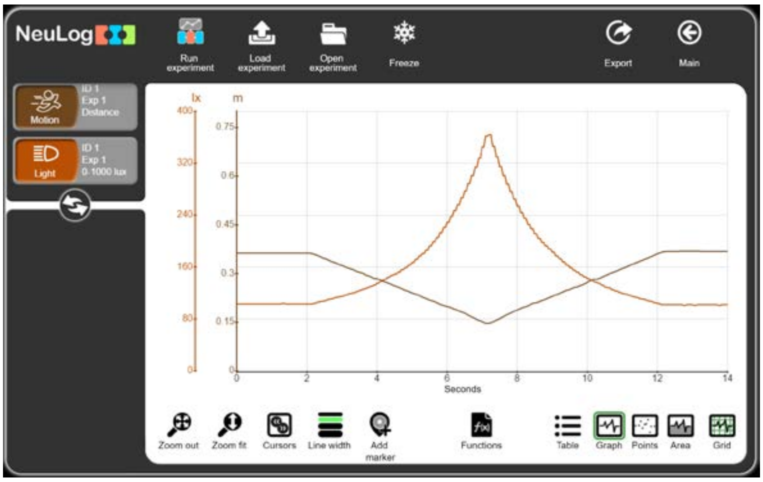
14. Run the Neulog software and wait for the following screen.




15. Click on the **Load experiment** button for the following screen.



16. Click on the **Load experiment** orange button for loading the experiment results from the sensors.




17. You can see on the graph the two seconds delays at the beginning and at the end, when the distance and the light are constant.

18. When the SENSE moves forward, the distance drops linearly and the light intensity increases in parabola shape.

19. When the SENSE moves backward, the distance increases linearly and the light intensity decreases in parabola shape.