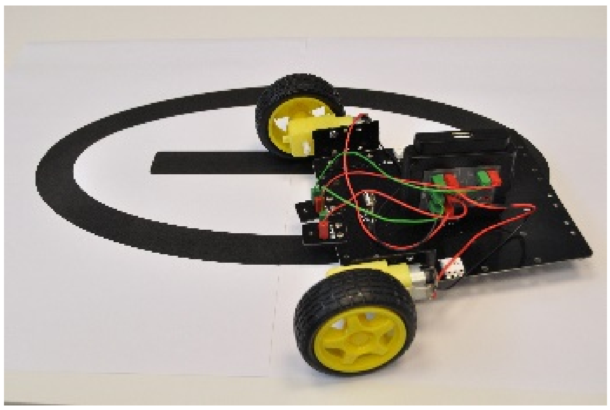
SENSE-MAKE
Robotics and Brain Units



NeuLog

SENSE-MAKE
Robotics and Brain Units

2_15

■ www.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. Few years ago, we could not understand 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.

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.

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.

Examples:

- 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".

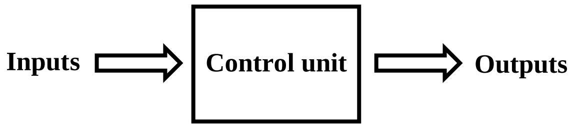


Figure 1-1

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 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 (ºC or ºF), 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.

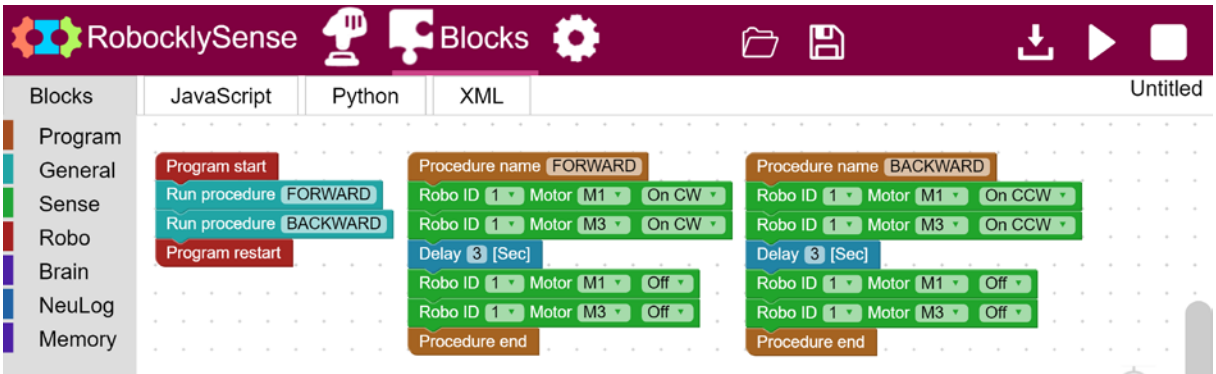
These brain units are connected in a chain to the main control unit, which communicates with them through messages.

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 for SENSE robots 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 ROBO-206

In this book, we will use the ROBO-206 as the control unit for computerized systems and robots.

The controller has three inputs (S1-S3) for connecting switches and sensors, three outputs (M1-M3) for the connection of lamps and motors, indicating LEDs, pushbutton switch and a mini USB connection for data transmission from the PC to the ROBO-206.

A program can be downloaded into the controller flash memory and it can be run using the pushbutton switch on its panel.

A battery module with USB outlet (like BAT-200 or power bank) can be connected to the ROBO-206 for operating the controller independently without a PC. Running and stopping the program is done by pressing the pushbutton on the panel.

In order to expand the number of inputs and outputs, an expansion unit (ROBO-216) can be connected in a chain to the ROBO-206.

Notes:

ROBO-206 is part of the NeuLog logger sensors family. These sensors can be connected to it and can be read by it.

The ROBO-206 can be also used as a USB adaptor for science experiments with the NeuLog software.

1.6 Preparing wires for the experiments

For executing the exercises described in the following experiments we need wires with plugs (wires and plugs can be found the kit).

Prepare the wires as follows:

a) Prepare 6 pairs of wires, 12cm long each.
b) Remove a short (5mm) section of the plastic insulation at the end of the wire.
c) Bend the ends of the wire strands back, so that they are flat against the insulation (see figure 1-2).
d) Insert the prepared wire end (with the strands folded over the insulation) into the plug end (see figure 1-2).
e) Tighten the screw. **Do not over tighten** so as not to cut the wire and break it.
f) The plug consists of three parts: the metal plug, a plastic insulator and a screw. If they are separated, insert the plug through the largest hole into the insulator. Note the direction of the insertion.
g) Use an additional plug to stabilize the metal plug in its insulator.
h) Screw the screw one turn into its thread (do not screw it all the way down).

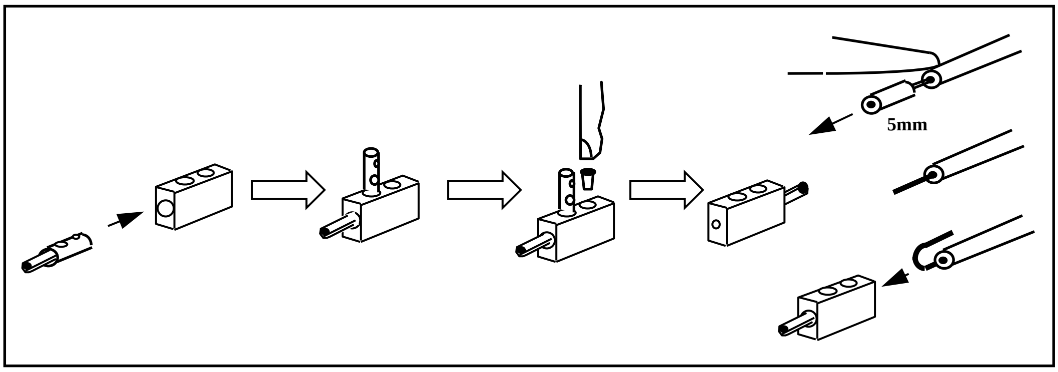


Figure 1-2

1.7 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 needed.

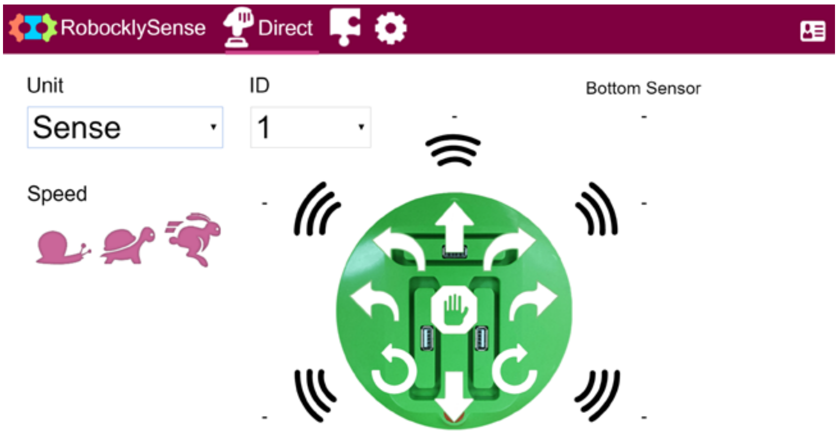
During upgrading the software the USB driver installation can be skipped by clicking the **Cancel** button.

1.8 Starting the RobocklySense program

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

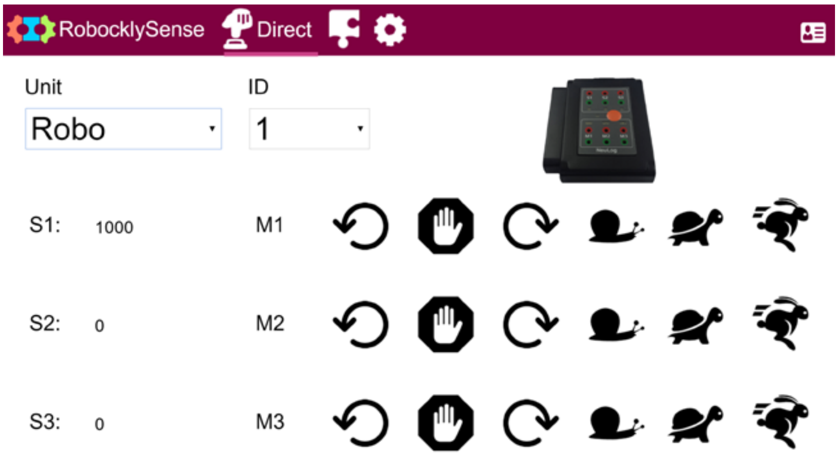
Click on the RobocklySense icon to run the RobocklySense software.

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



This screen is the **Direct** mode for the SENSE robot.

Change the **Unit** field to **Robo** and you will get the following screen:



This screen is 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-MAKE units and components.
- How to connect components to the ROBO-206.
- How to operate the units at direct mode.

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE components kit

Discussion:

The RobocklySense software enables to operate the ROBO-206 controller directly.

We will connect the kit modules to the ROBO-206 and learn how to operate them directly without programming.

Procedure:

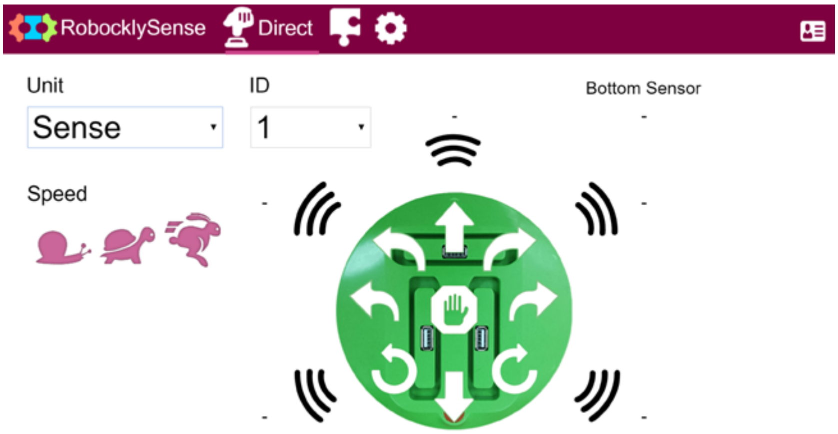
1. The SENSE-MAKE kit includes the following components:

Each electrical component is connected to the ROBO-206 controller by two electrical wires with plugs.

2. Connect the ROBO-206 controller to the PC using the USB cable.

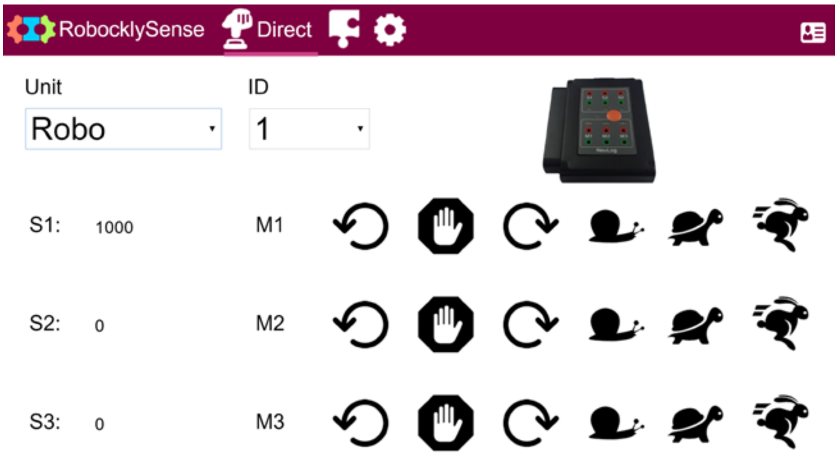
3. Run the RobocklySense software.

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



This screen is the **Direct** mode for the SENSE robot.

4. Change the **Unit** field to **Robo** and you will get the following screen:



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.

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 Motor test

1. Connect one of the motors to the M1 terminals.

2. Click on the **ON CW**  icon on M1 line.

   The motor should rotate clockwise.

3. Click on the **OFF**  icon and check that the motor stops.

4. If the motor rotates counterclockwise, change the wires connected to the motor between them and check the motor rotation.

5. Click on the **ON CCW**  icon.

   The motor should rotate counterclockwise.

6. Change the rotation direction of the motor using the direction commands: and .

   <u>Note:</u>
   Changing direction without first stopping the motor, shorten its life. It is recommended to stop the motor before changing its direction.

7. Change the rotation speed of the motors using the speed commands:

8. Click on the **OFF** icon and check that the motor stops.

1.1.2 Buzzer test

1. Connect the buzzer to the M1 terminals instead of the motor.

   We use the motor unit to control the buzzer.

2. Click on the **ON CW** icon on M1 line.

   The buzzer should turn ON.

   If not, change the connection of the wires.

   In order for the buzzer to operate, the current through the buzzer should flow in a certain direction.

3. Click on the **OFF** icon and check that the buzzer is OFF.

4. Click on the **ON CCW** icon.

   The buzzer **should not** turn ON.

   The output current direction affects the buzzer.

5. Click on the **OFF** icon.

1.1.3 LEDs test

The LEDs unit includes 3 LEDs (red, yellow and green).

1. Connect the red LED to the M1 terminals.

2. Connect the yellow LED to the M2 terminals.

3. Connect the green LED to the M3 terminals.

   We will use the motor unit to control the LEDs.

4. Click on the **ON CW** icon of **M1**.

   The red LED should turn ON.

   If not, change the connection of the wires.

   In order to light the LED, the current through the LED should flow in a certain direction.

5. Click on the **OFF** icon and check that the LED is OFF.

6. Click on the **ON CCW** icon.

   The LED **should not** turn ON.

   The output current direction affects the LED.

7. Click on the **OFF** icon.

8. Click on the **ON CW** icon of **M2**.

   The yellow LED should turn ON.

9. Click on the **OFF** icon and check that the LED is OFF.

10. Click on the **ON CW** icon of **M3**.

    The green LED should turn ON.

11. Click on the **OFF** icon and check that the LED is OFF.

12. Disconnect the LEDs.

1.1.4 Lamp test

1. Connect the lamp to the M1 terminals.

   We will use the motor unit to control the lamp.

2. Click on the **ON CW** icon of **M1**.

   The lamp should turn ON.

   If not, change the connection of the wires.

   The lamp is also a LED. In order to light, the current should flow in a certain direction.

3. Click on the **OFF** icon and check that the lamp is OFF.

4. Click on the **ON CCW** icon.

   The lamp **should not** turn ON.

   The output current direction affects the lamp.

5. Click on the **OFF** icon.

6. Do not disconnect the lamp.

1.1.5 Light sensor test

1. Check that the lamp is OFF.

2. Connect the terminals of a **Light** sensor to the **S1** terminals.

   The read value should appear in the **S1** field.

   The sensor reads the surrounding light.

3. If the value is 0, check that the sensor wires are right connected.

4. Turn the lamp ON and put it against the light sensor.

   The read value should be much higher than was before.

   It takes about 1 second until the value is changed.

5. Block the lamp's light.

   The read value should change.

6. Click on the **OFF**  icon to turn the lamp OFF

7. Disconnect the lamp and the light sensor.

1.1.6 Switch test

1. Connect the switch to the **S1** terminals.

2. Move the switch to the right.

   The read value should be 0.

3. Move the switch to the left.

   The read value should be 1000 (the maximum value).

   When the switch is on the left, the input terminal is connected to 5V, and this voltage is converted to the number 1000.

1.1.7 Tactile switch test

1. Connect one of the Tactile Switch (Tact) to the **S1** terminals.

   The read value should be 0.

2. Press the switch and keep it pressed.

   The read value should be 1000 (the maximum value).

   When the switch is pressed, the input terminal is connected to 5V, and this voltage is converted to the number 1000.

3. Close the RobocklySense software.

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-MAKE components kit

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 Robockly 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 LEDs module to the base card as in the following picture:

2. Install the ROBO-206 on the base card.

3. Connect the ROBO-206 to the PC using the USB cable.

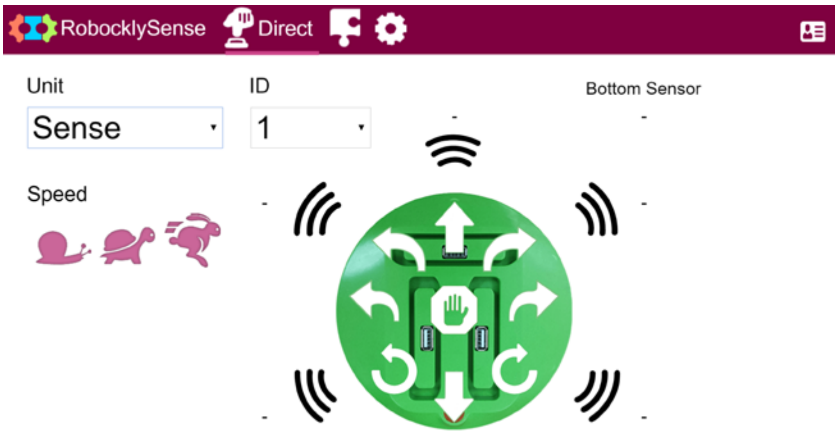
4. Connect the red LED to the M1 terminals.

5. Connect the yellow LED to the M2 terminals.

6. Connect the green LED to the M3 terminals.

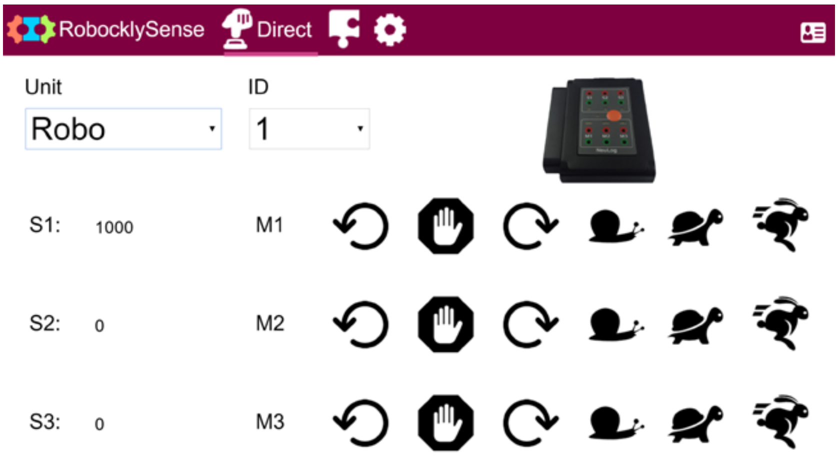
7. Run the RobocklySense software.

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



This screen is the **Direct** mode for the SENSE robot.

8. Change the **Unit** field to **Robo** and you will get the following screen:

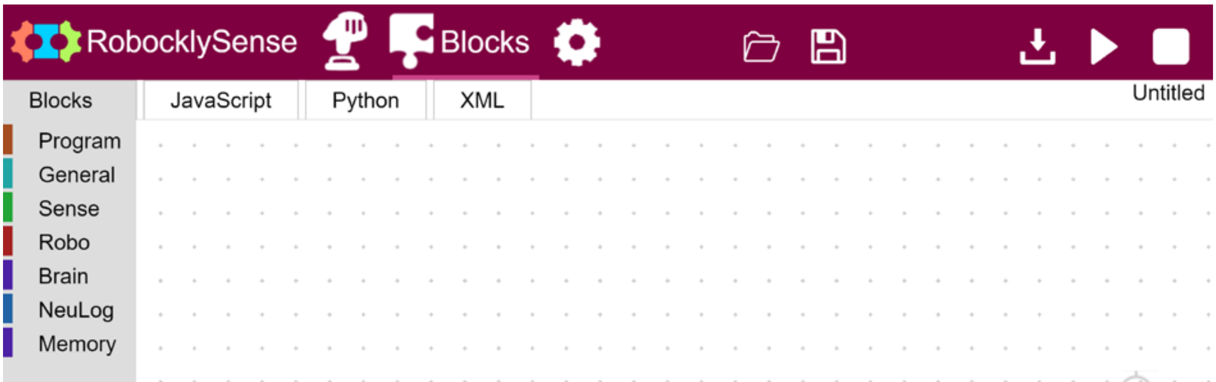


The direct mode enables you to test the system and its wiring before programming and running the programs. This stage saves a lot of frustration in development.

9. Test the three LEDs as described in experiment 1.1.

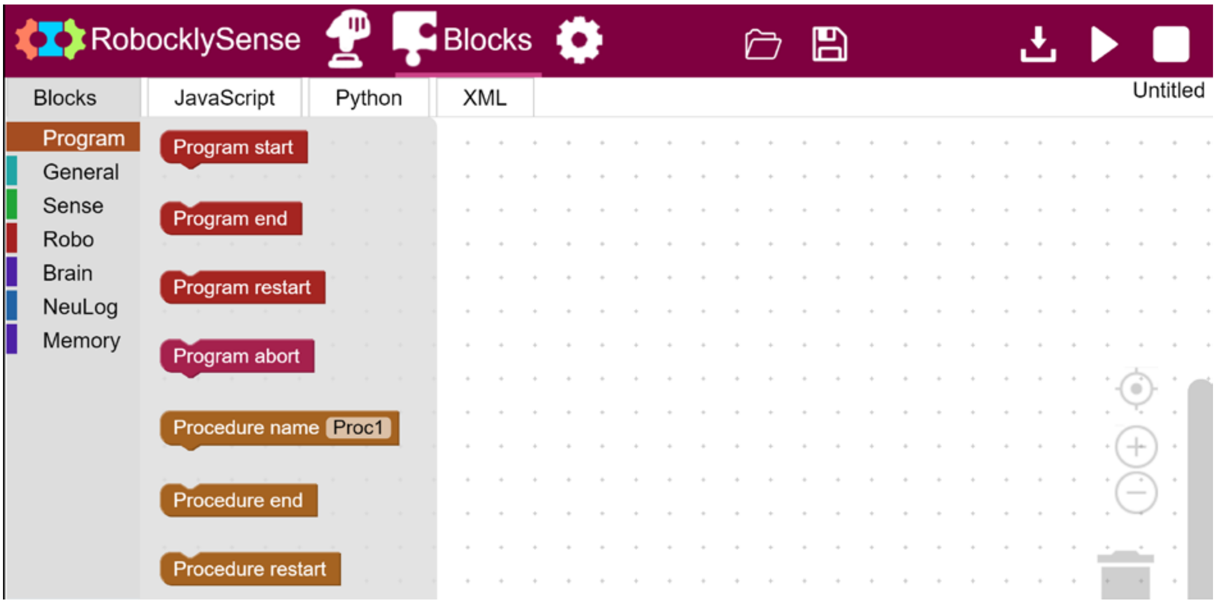
1.2.1 First program – blinking LED

1. Move to **Block** mode.

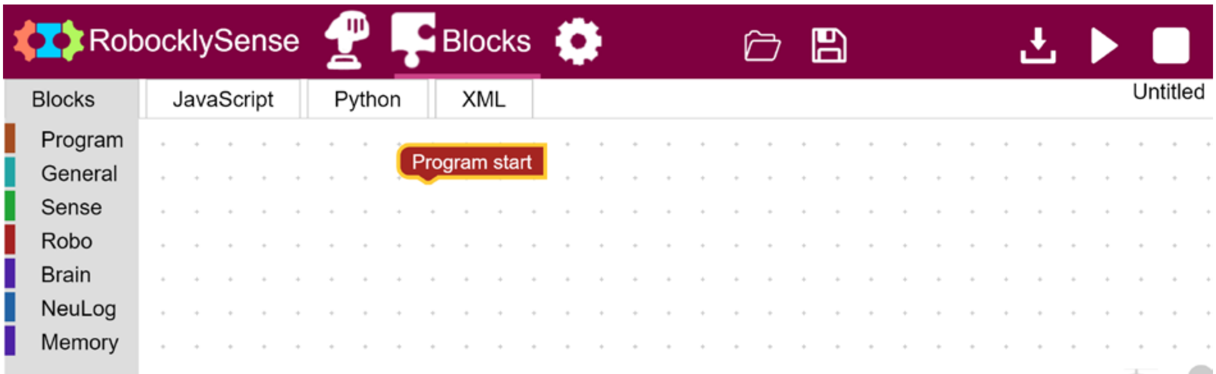


A computer program is composed of chains of instructions.

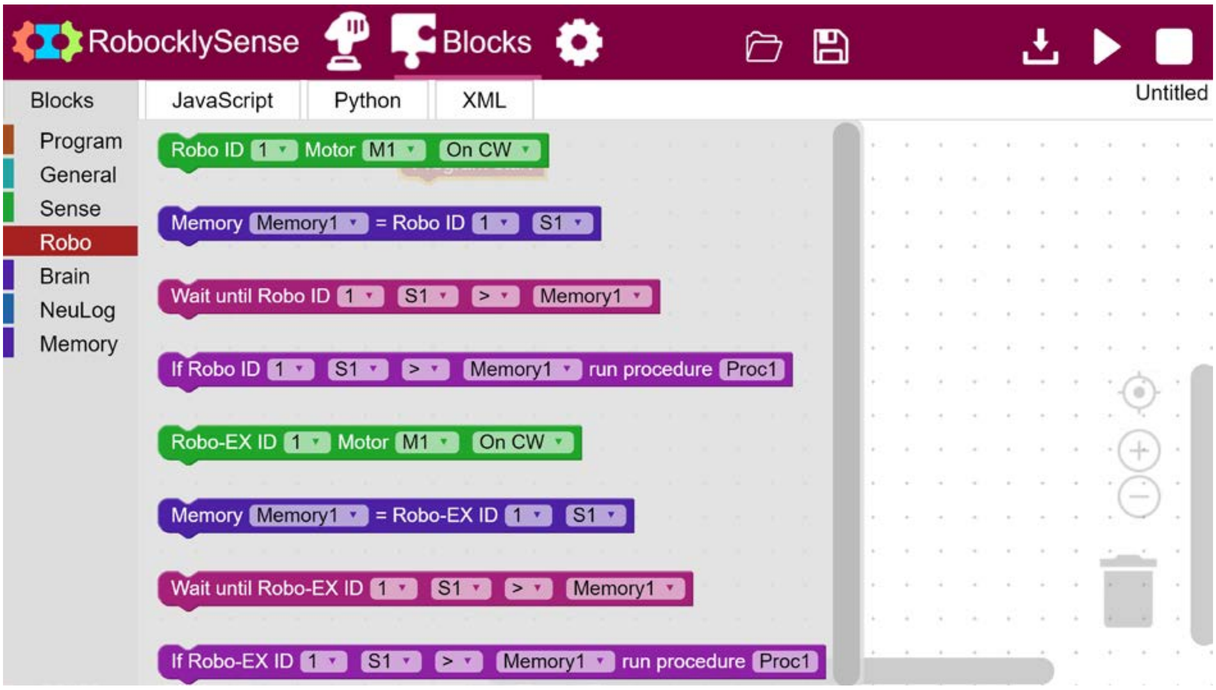
2. Click on the **Program** button and a list of program instruction list will appear:



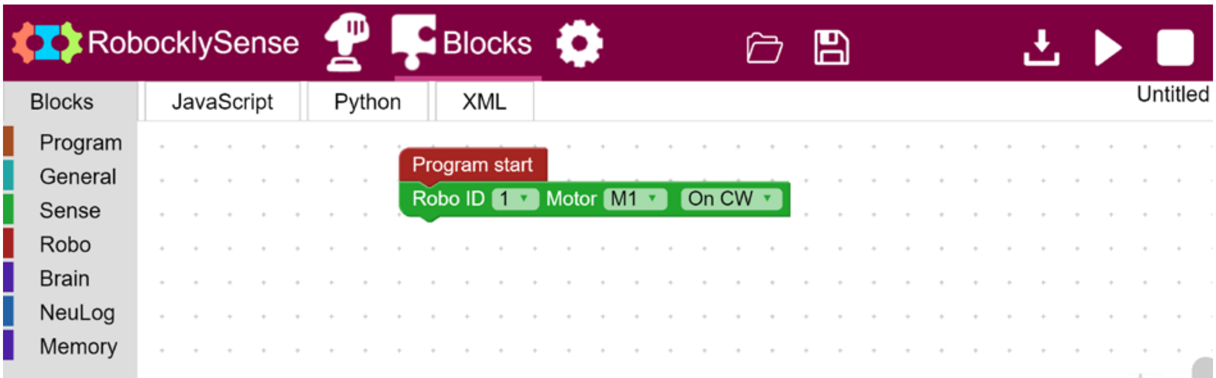
3. Click on the **Program start** instruction block and drag it to the right.



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

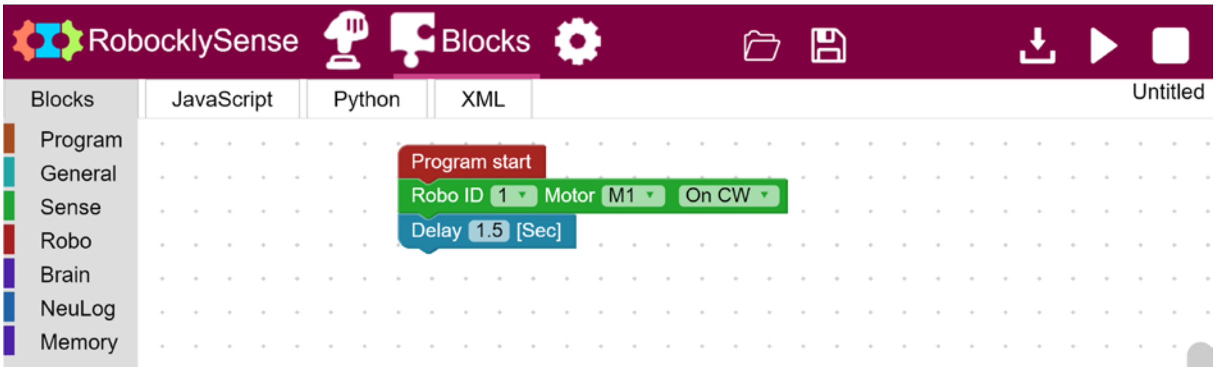


5. Click on the **Robo Motor** instruction block and drag it under the **Program start** instruction block. They will be attached together.



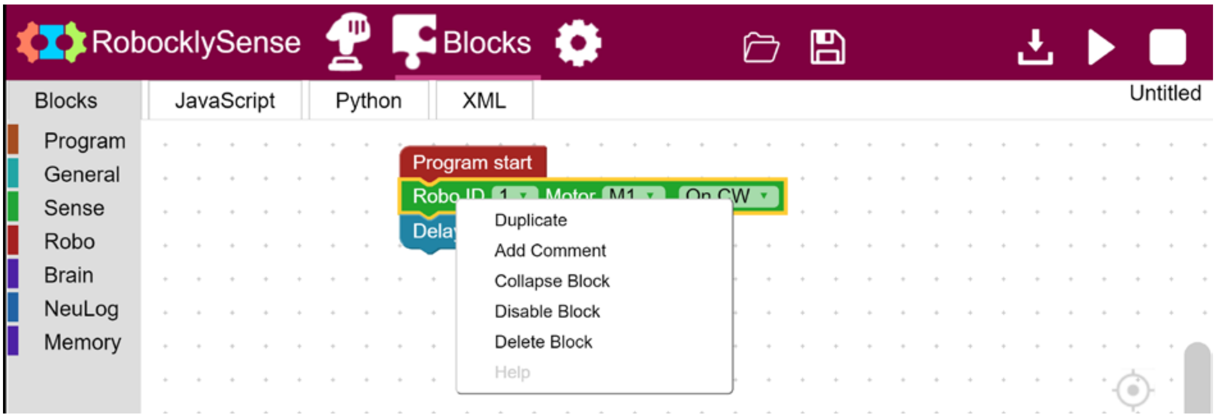
6. Click on the **General** button and select the **Delay** instruction block.

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



7. Change the delay value to 5 seconds.

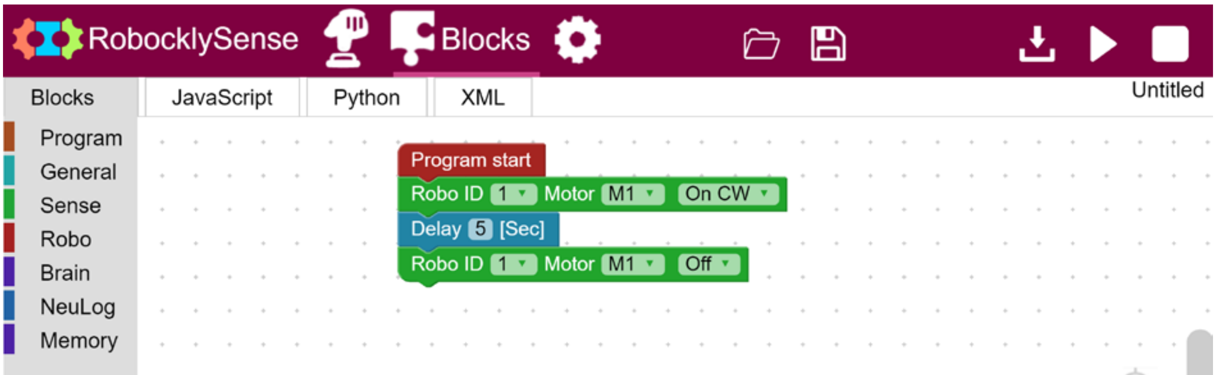
8. Right click on the **Motor** instruction block.



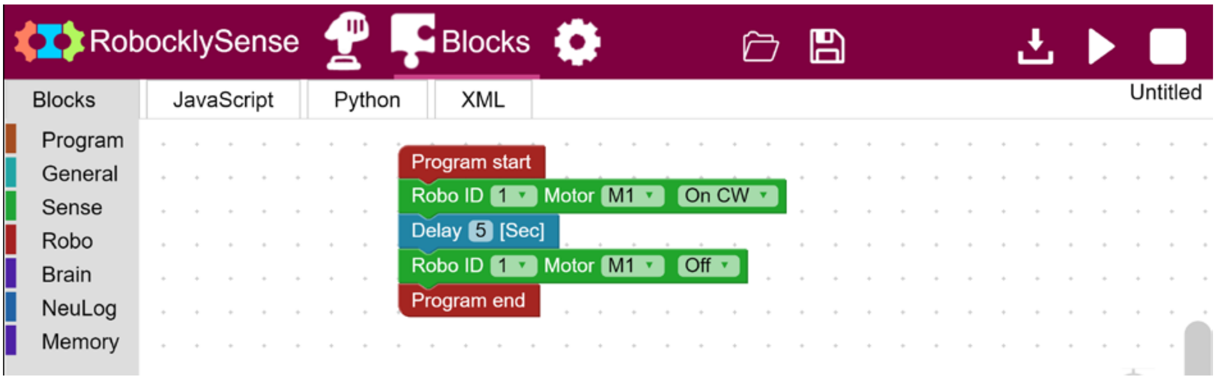
9. Select **Duplicate** and a new **Motor** instruction block will appear. Attach it to the **Delay** instruction block.

10. Change the command of the new **Motor** instruction block to **Off**.

11. Check that you have the following program:



12. Click on the **Program** button and select the **Program end** instruction block. Drag it under the last **Motor** instruction block.



13. Click on the **Save**  button and save the program under the name **LAMP1**.

1.2.2 Program download

1. The ROBO-206 is also a computer with a flash memory.

   We can download the program into its memory.

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

   This will transfer the PC program into the ROBO-206 flash memory and replaces a previous program, if exists in it.

3. Click on the **Run** button.

   The lamp will turn ON for 5 seconds and then OFF.

4. To run the program in the controller memory, we can also use the **Start/Stop** pushbutton located on the controller panel.

   Press the ROBO-206 panel pushbutton and you will see the lamp blinking once.

   You can close the RobocklySense software and run the program using this pushbutton.

   In this case, the PC is used only as a power supply.

   Replacing the PC by a battery module create an independent robotic controller.

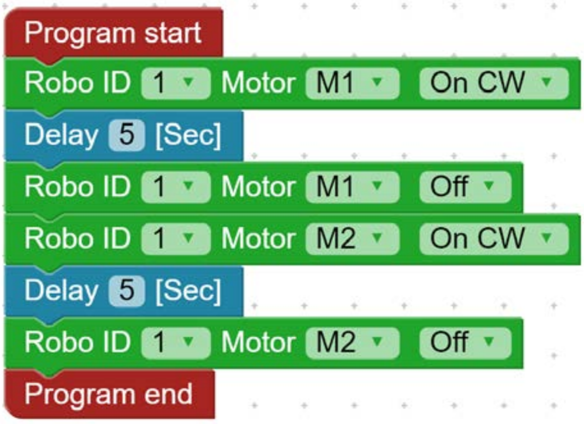
   <u>**Important notes:**</u>

   Reconnecting the ROBO-206 to the PC should be done when it is not running (the panel LED is not blinking). You can leave the battery Power Bank connected to the ROBO-206 socket.

1.2.3 Blinking LEDs

1. Duplicate the **Delay** and the two **Motor** instruction blocks.

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



3. Click on the **Save** button and save the program under the name **LAMP2**.

4. Click on the **Program download** button.

5. Click on the **Run** button.

   The green lamp will turn ON for 5 seconds, turn OFF and the red lamp turns ON for 5 seconds and then OFF.

1.2.4 Challenge exercises – Blinking LEDs

Task 1: Improve the LAMP2 program so all the three LEDs will be ON together for 3 seconds and OFF together.

Download the program into the controller and run it.

Run the program from the computer and check its behavior.

Task 2: Change the program so the LEDs will be ON in a sequence of one at a time.

Download the program into the controller and run it.

Run the program from the computer and check its behavior.

Experiment 1.3 – Procedures as New Instructions

Objectives:

- Using procedures in a program.
- Building and implementing a traffic light.

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE components kit

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.

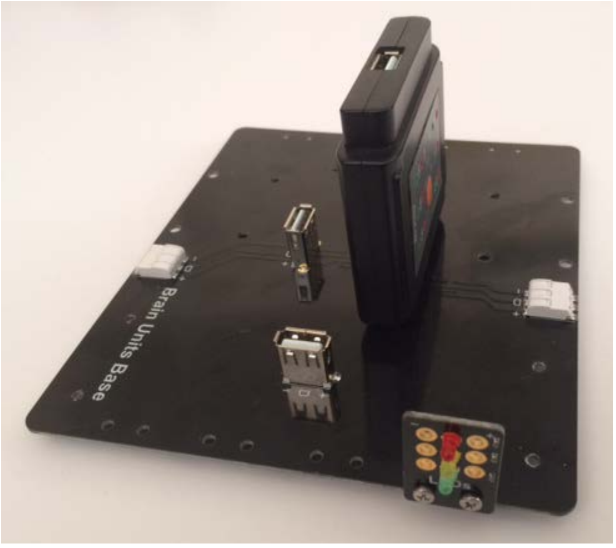
In a program, there is one main program and procedures. In this way, when we run the program, the computer knows where to start.

In this experiment, we create a program that blinks the lamps when a switch is pressed. We shall use a procedure that blinks the lamps.

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 LEDs module to the base card as in the following picture:



2. Install the ROBO-206 on the base card.

3. Connect the ROBO-206 to the PC using the USB cable.

4. Connect the red LED to the M1 terminals.

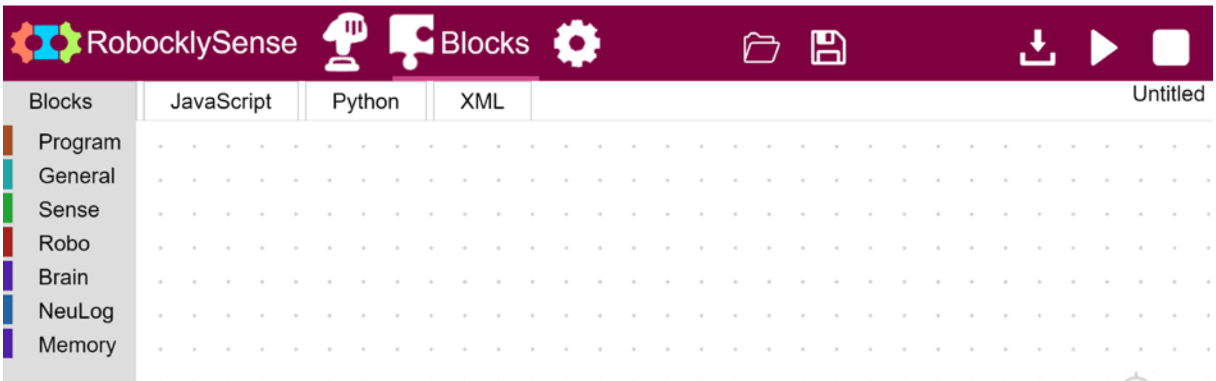
5. Connect the yellow LED to the M2 terminals.

6. Connect the green LED to the M3 terminals.

7. Run the RobocklySense software.

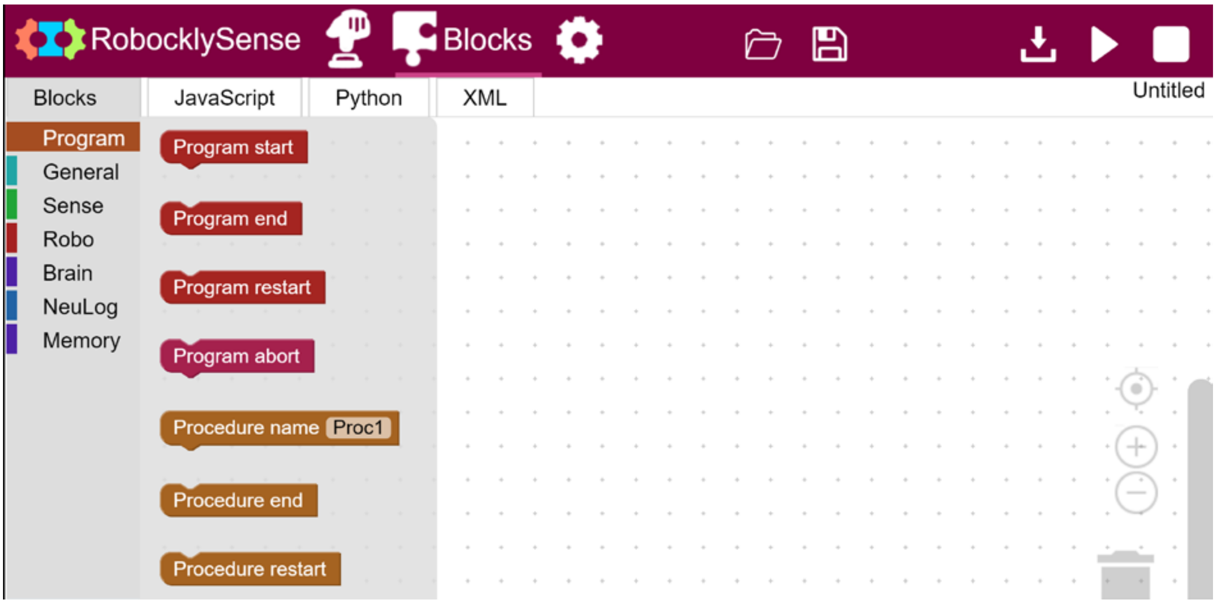
8. Test the three LEDs as described in experiment 1.1.

9. Move to **Block** mode.

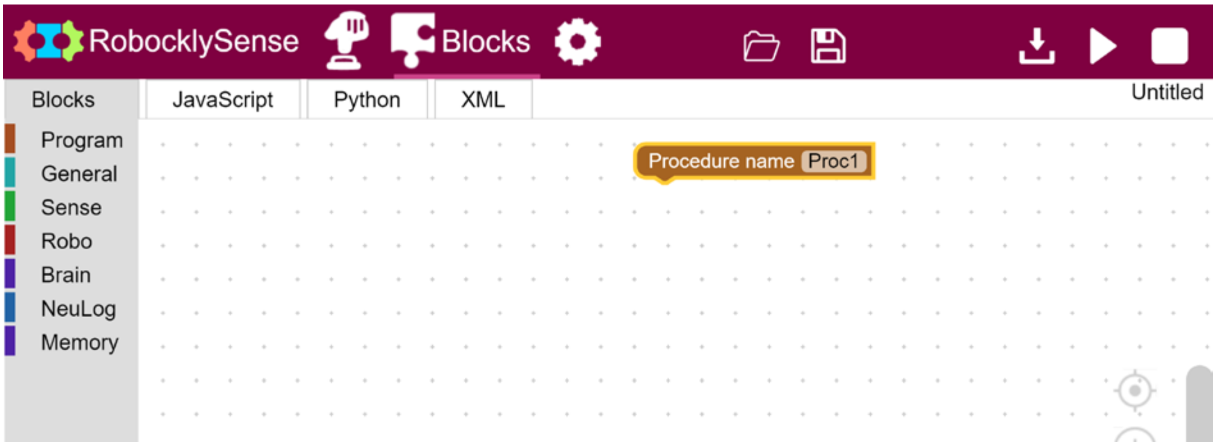


1.3.1 Programs and procedures

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



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



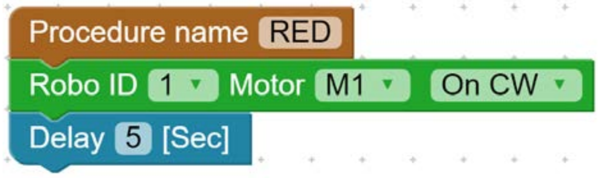
3. The **Procedure name** instruction block has a field for the procedure name.

   Write the name **RED** in this field.

4. Add the **Robo Motor** instruction under the **Procedure Name** instruction block.

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

   Drag it under the **Motor** instruction block and change the value to 5 seconds.

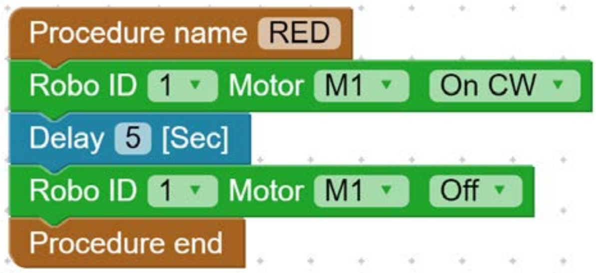


6. Duplicate the **Motor** instruction and attach it to the **Delay** instruction block.

7. Change the command of the new **Motor** instruction block to **Off**.

8. Click on the **Program** button and select the **Procedure end** instruction block.

   Drag it under the last **Motor** instruction block.



9. Repeat above steps and create the **YELLOW** procedure with a 2 seconds delay.

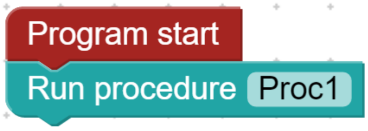
10. Repeat above steps and create the **GREEN** procedure with a 4 seconds delay.

11. We shall build now the main program that calls these procedures.

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

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

13. Click on the **General** button and add the instruction **'Run procedure Proc1'**.

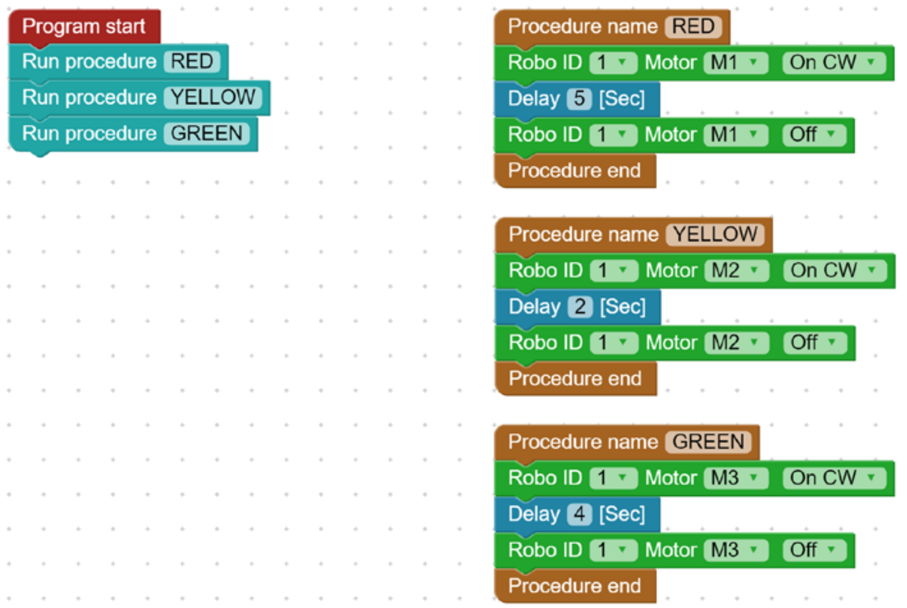


14. Change the name **Proc1** to **RED**.

15. Create the instruction **'Run procedure YELLOW'**.

16. Create the instruction **'Run procedure GREEN'**.

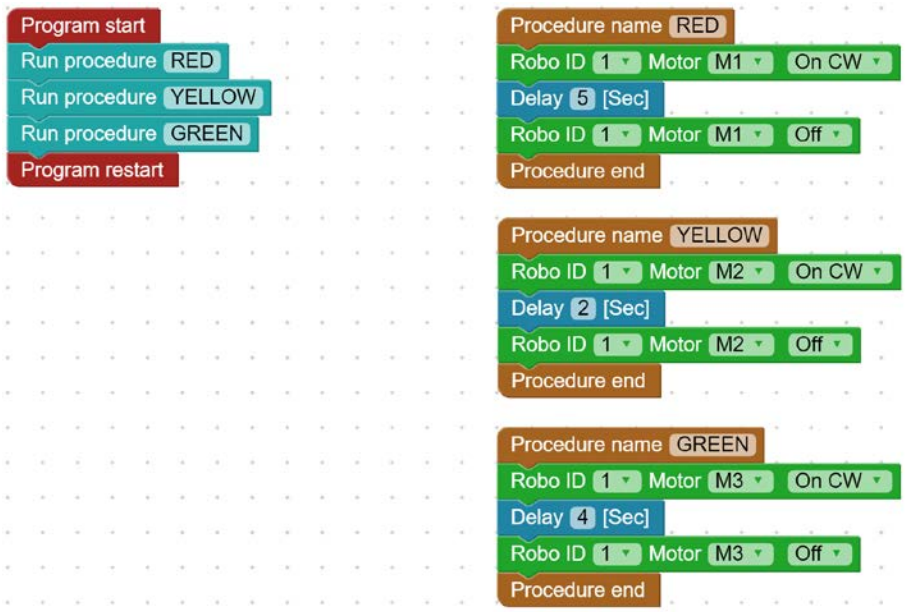
17. Check that you have the following program:



1.3.2 Endless loop

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

1. Add the instruction **Program restart** under the **RUN procedure** instructions.
2. Check that you have the following program:



3. Click on the **Save** button and save the program under the name **Lights2**.

4. Click on the **Program download** button.

5. Click on the **Run** button.

   The red LED on the ROBO-206 panel should blink indicating that its program is running.

   The three LEDs will light in a sequence of traffic lights in endless loop.

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

6. Stop the program by clicking the **Stop** button.

7. To run the program in the ROBO-206 memory, we can also use the **Start/Stop** orange pushbutton located on the ROBO-206 panel.

   Press it and the three LEDs will light in a sequence of traffic lights in endless loop.

8. To stop the system running, press the **Start/Stop** pushbutton switch on the ROBO-206. The red LED on its panel should stop blinking.

   You may have the NeuLog battery module BAT-202, which can be plugged directly into the ROBO-206 or to one of the base sockets.

   When connecting such battery to the ROBO-206 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 base socket and disconnect the ROBO-206 from the PC.

The menu line is changed to the following:

9. Press the **Start/Stop** button and the three LEDs will light in a sequence of traffic lights in endless loop.

10. To stop the system running, press the **Start/Stop** pushbutton switch on the ROBO-206. The red LED on its panel should stop blinking.

11. Connect again the ROBO-206 to the PC and wait until the menu line is changed back to the following.

1.3.3 Challenge exercises – Traffic lights

Task 1: Increase the delay time of the red LED and decrease the delay time of green LED.

Notice that with procedures you do not have to search for the relevant delay in the main program.

Download the program, run and check it.

Task 2: Change the **Lights2** program so the green LED blinks twice (one second ON and One second OFF) before turning OFF.

Download the program, run and check it.

Experiment 1.4 – Switches

Objectives:

- Responding to switches.

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE components kit

Discussion:

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).

Of course, we need to take care that the turning on light does not affect the light sensor.

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.

1.4.1 Memories and variables

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

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.

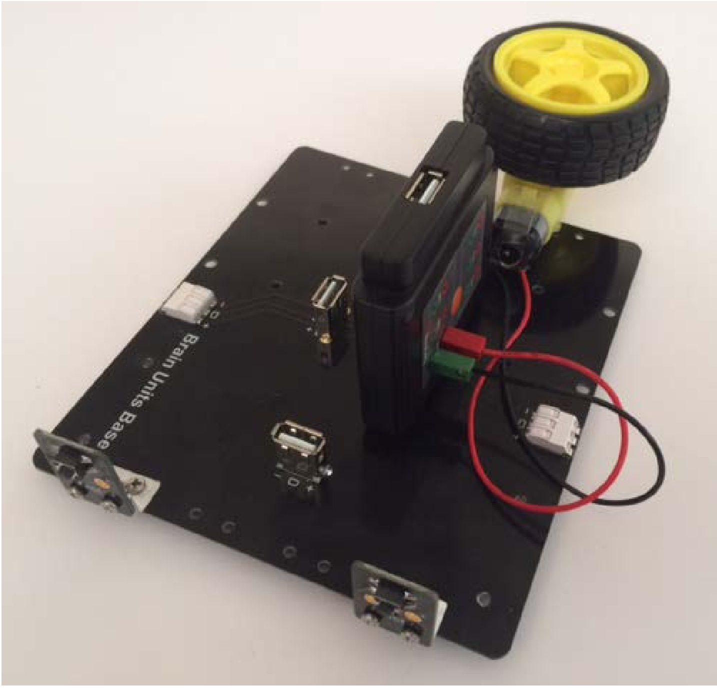
A **variable** is a memory cell with a name. In the instruction we indicate the name of the variable.

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

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

Procedure:

1. Connect the two **Tact Switch** modules and the **Motor** to the base card as shown in the following picture:



2. Install the ROBO-206 on the base card.

3. Connect the ROBO-206 to the PC using the USB cable.

4. Connect the motor to the M1 terminals.

5. Connect the left tact switch to the S1 terminals.

6. Connect the right tact switch to the S2 terminals.

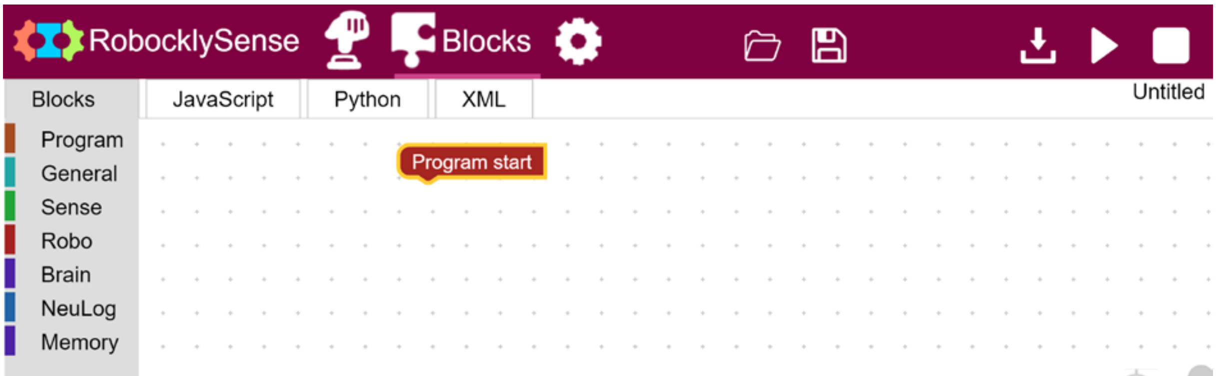
7. Run the RobocklySense software.

8. Test the motor as described in experiment 1.1.

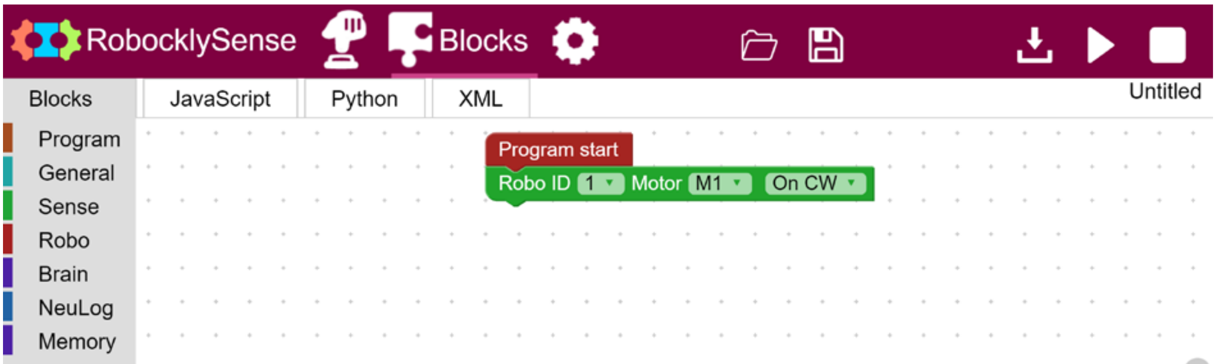
9. Test the tact switches as described in experiment 1.1.

10. Move to **Block**  mode.

11. Click on the **Program** button, click on the **Program start** instruction block and drag it to the right.



12. Click on the **Robo** button and add the **Robo Motor** instruction block under the **Program start** instruction block.



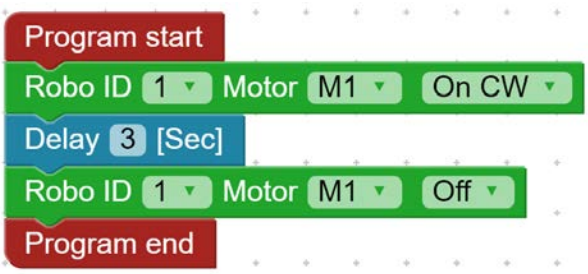
13. Click on the **General** button and add the **Delay** instruction block.

14. Change the delay value to 3 seconds.

15. Duplicate the **Motor** instruction block. Attach it to the **Delay** instruction block.

16. Change the command of the new **Motor** instruction block to **Off**.

17. Click on the **Program** button and select the **Program end** instruction block. Drag it under the last **Motor** instruction block.



18. Click on the **Save** button and save the program under the name **Motor1**.

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

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

    The motor will turn ON for 3 seconds and then OFF.

21. Before changing the motor direction, it is better to stop it for short time and then turning it ON to the other direction.

    Change the program to the following one:

22. Download, run and check the program behavior.

23. Replace the **Program end** instruction with the **Program restart**.

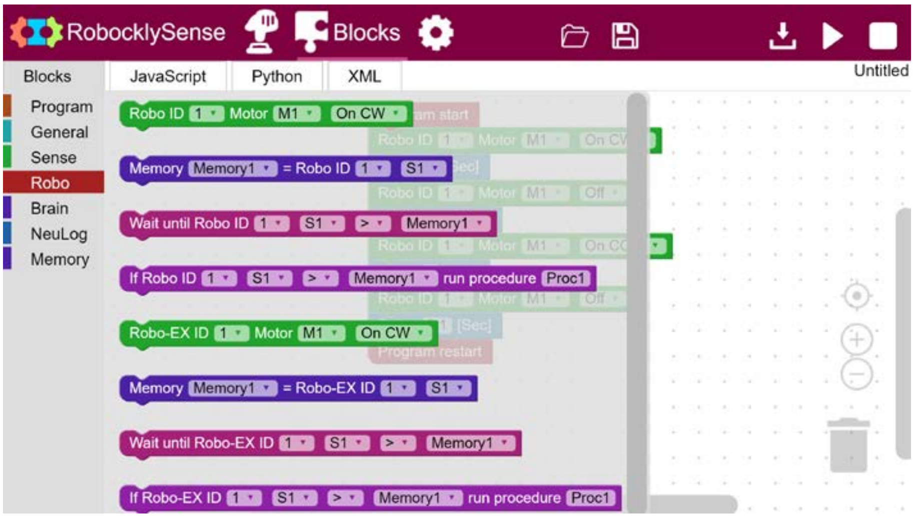
24. Download, run and check the program behavior.

25. Stop the program running.

1.4.2 Wait until

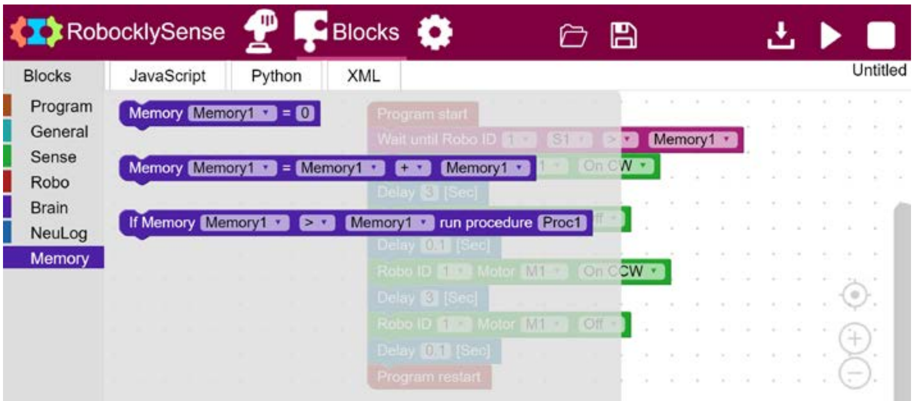
We shall add the instruction **Wait until** at the beginning of the program.

1. Click on the **Robo** button and a list of **Robo** instructions appears.



2. Click on the **'Wait until ID 1 S1 > [memory1]'** instruction block and drag it to be the first instruction in the program.

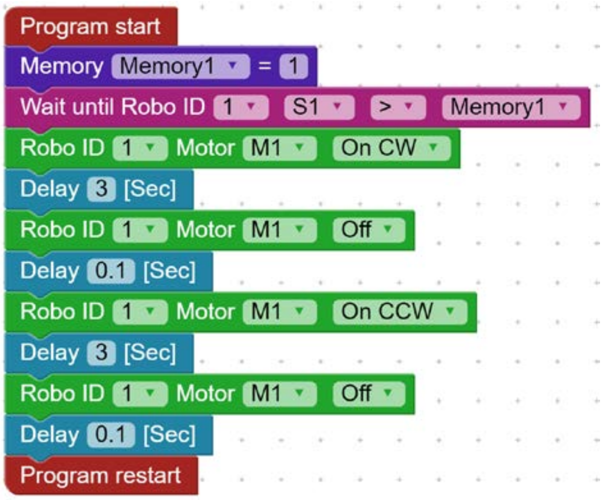
3. Click on the **Memory** button and a list of input instructions appears:



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

5. Change the number in the instruction to 1.

6. Check that you have the following program:



7. Save the program under the name **SWITCH1**.

8. Download and run the program.

   Nothing will happen to the motor.

9. Press the tact switch.

   The motor will run and change rotation direction as long as you press the tact switch.

   <u>**Note:**</u>
   As described in experiment 1, when the switch is pressed, the read value in S1 is 1000.

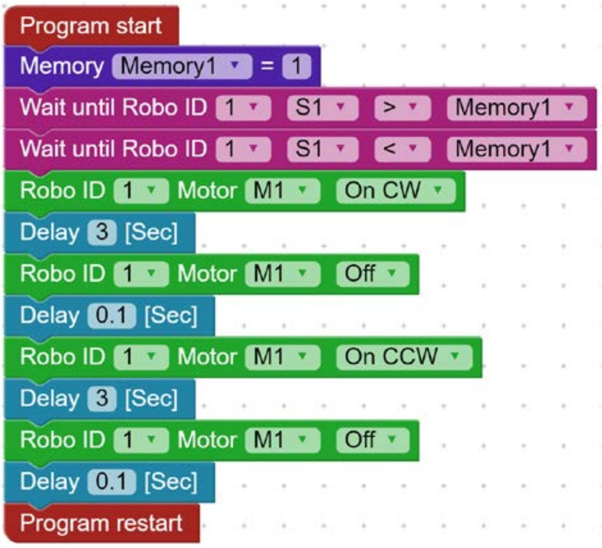
10. Release the tact switch.

    The motor will stop after finishing the cycle.

    The program waits for the switch.

11. Stop the program's running.

12. Change the program so that each pressing and releasing of the tact switch will cause one cycle of the motor.



13. Save the program.

14. Download and run the program.

    The motor will do its cycle once only after pressing and releasing the tact switch.

    Each pressing and releasing will cause one cycle of the motor.

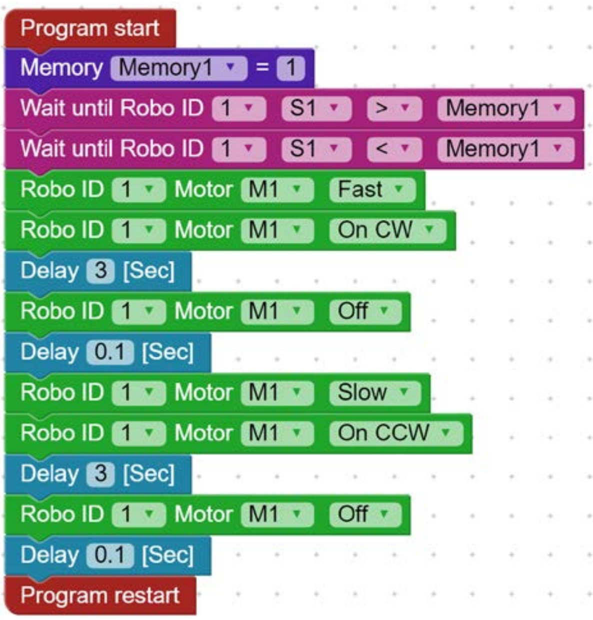
    Check this.

1.4.3 Changing speeds

The **Motor** command enables us to change the motor speed too.

After changing speed, the speed is set until the next change.

1. Change the program so that each pressing and releasing of the tact switch will cause one cycle of the motor. The motor speed is fast CW and slow CCW.



2. Save the program.

3. Download and run the program.

   The motor will do its cycle once only after pressing and releasing the tact switch with different speeds.

   Each pressing and releasing will cause one cycle of the motor.

   Check this.

1.4.4 Challenge exercises – Motor controlled by tact switch

Task 1: Create a program that turns the motor ON after press and release of the tact switch and turns it OFF after another press and release of the tact switch.

The program runs in an endless loop.

Download, run and check the program.

Task 2: Create a program that turns the motor ON after three presses of the left tact switch and turns it OFF after one press and release of the right tact switch.

The program runs in an endless loop.

Download, run and check the program.

Experiment 1.5 – Conditions and Decisions

Objectives:

- The If instruction.
- OR condition.
- AND condition.

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE components kit

Discussion:

1.5.1 The If instruction

The condition instructions are conditional execution instructions.

The instruction is executed when the condition written in the instruction exists.

The conditional instruction in the RobocklySense software must call a procedure.

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

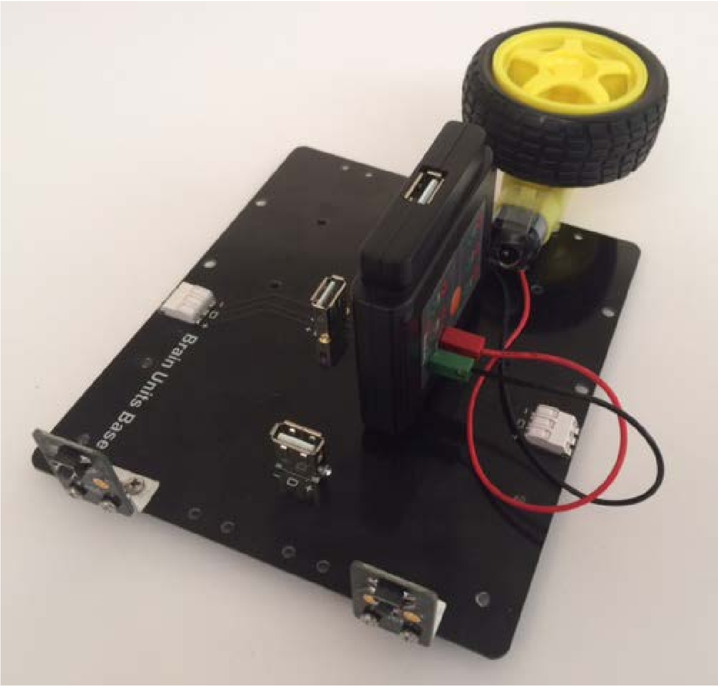
The condition is checked when the condition instruction is executed. If the condition is not exists, the program continues to the following instruction.

The following program is a program for movement along a black line using **If** instructions.

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

Procedure:

1. Connect the two **Tact Switch** modules and the **Motor** to the base card as shown in the following picture:



2. Install the ROBO-206 on the base card.

3. Connect the ROBO-206 to the PC using the USB cable.

4. Connect the Motor to the M1 terminals.

5. Connect the left Tact Switch to the S1 terminals.

6. Connect the right Tact Switch to the S2 terminals.

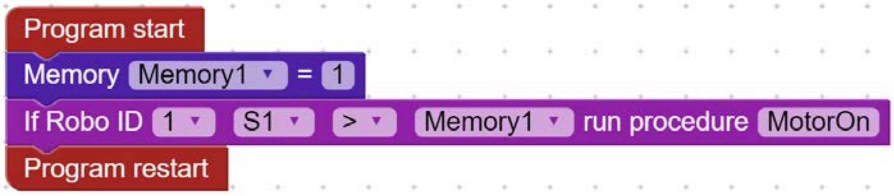
7. Run the RobocklySense software.

8. Test the tact switches as described in experiment 1.1.

9. Test the motor as described in experiment 1.1.

10. Create the **MotorOn** procedure with a single instruction:

11. Create the main program with the following instructions:



12. Save this program under the name **MOTOR1**.

13. Download and run the program.

    Nothing will happen.

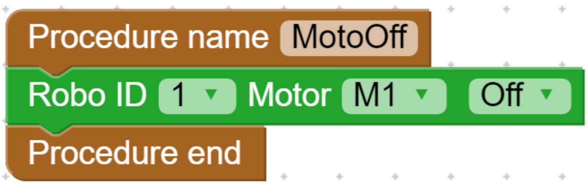
14. Press the tact switch.

    The motor will turn ON and will not turn OFF again.

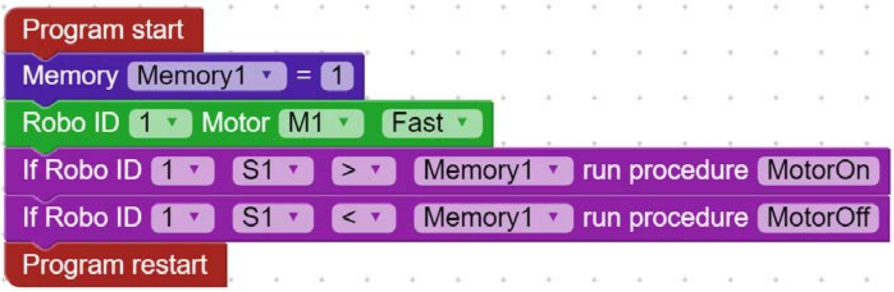
15. Stop the program and the motor.

    To turn the motor OFF, we need to give the turning **Off** instruction.

16. Create **MotorOff** procedure with a single instruction:



17. Change the program to the following one:



18. Download and run the program.

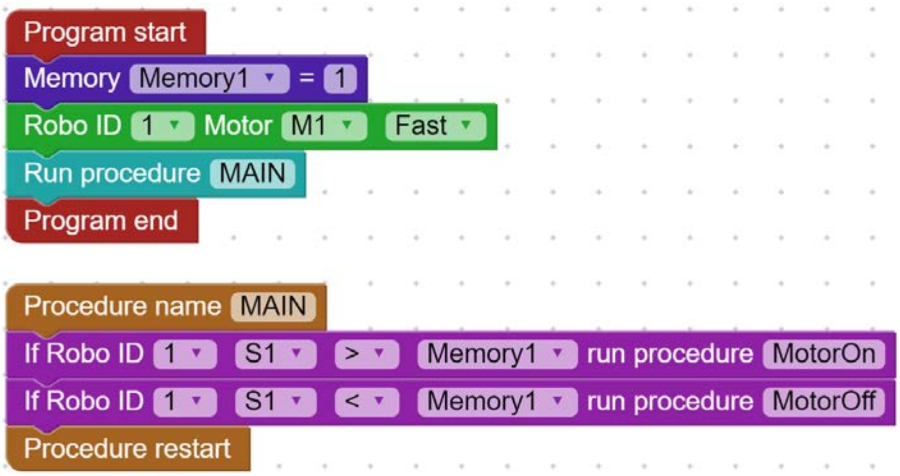
    This time, pressing the tact switch will turn the motor ON and releasing the tact switch will turn the motor OFF.

1.5.2 Definitions

The definition of Var1 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.

1. Change the program to the following program and **MAIN** procedure:



2. Download and run the program.

   Pressing the tact switch will turn the motor ON and releasing the tact switch will turn the motor OFF.

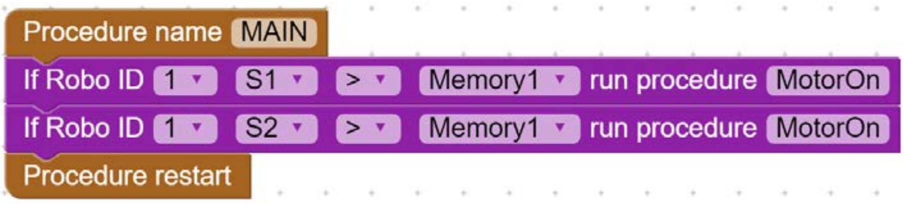
   Var1 is defined only once in the program.

1.5.3 OR condition

We would like to operate the motor by the two tact switches according to OR condition.

In OR condition the motor is ON when the left tact switch is ON **or** the right tact switch is ON **or** both of them are ON.

1. Change the main procedure to the following one:



2. Download and run the program.

3. Press one of the tact switches and check that the motor turns ON.

4. Stop the program running.

   The motor should stop.

   If not, stop it in **Direct** mode.

5. Run the program again.

6. Press the other tact switch and check that the motor turns ON.

7. Stop the program.

1.5.4 AND condition

To stop the motor we need the AND condition.

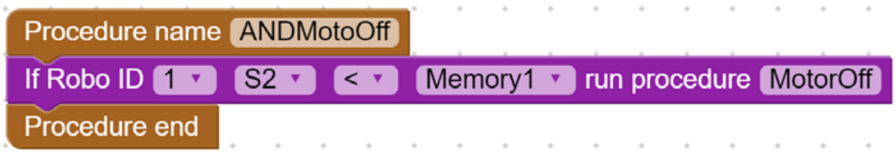
The motor should be OFF when the two switches are OFF.

In some programming software, the **AND** condition can be added to the **IF** instruction.

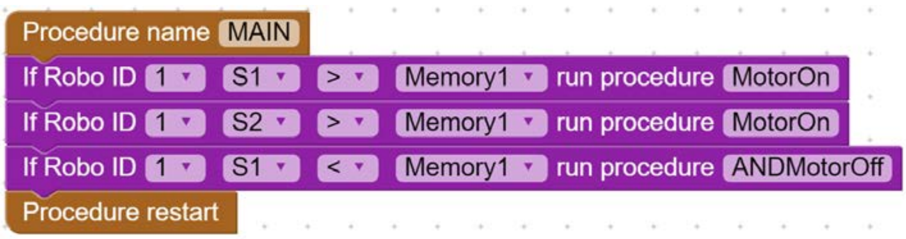
The RobocklySense does not have this option.

The AND operation can be achieved by creating another procedure.

1. Create the **AND MotorOff** procedure with single instruction:



2. Change the main program to the following one:



3. Analyze the program and the procedures.

4. Download and run the program.

5. Press the tact switches and check if the motor is OFF only when the two switches are OFF

1.5.5 Challenge exercise – Motor controlled by two tact switches

Task 1: Build a program that turns the motor ON when the left tact switch is pressed and released and turns OFF when the right tact switch is pressed and released.

Experiment 1.6 – Light Gate

Objectives:

- How brain units work?
- How Servo motors?
- Brain servo motor.
- Light sensor.
- Optical coupler.

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE components kit

Discussion:

1.6.1 Brain units

The brain unit is a unit with its own controller that takes care of the controlled component. For example, a unit that controls the motor's speed and direction, lamp intensity, servo motor angle, reading a sensor, etc.

Three wires connect the brain unit to the system control unit. The brain unit has two outlets of three wires each.

The 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. The messages from the control unit start with ID number. Only the brain unit with this ID number interprets the message and executes it.

This way of system construction is the way modern systems are built and have important advantages.

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.

This kind of system can easily be changed and expanded and not depends on the control units number of inputs and outputs.

1.6.2 Brain servo motor

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

The servo motor of the SENSE-MAKE kit has transmission gear and potentiometer. The potentiometer is variable resistors that create variable voltage according to the servomotor shaft angle.

The brain servo motor controller 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.

The kit contains two brain servomotors – one with ID #1 and one with ID #2.

1.6.3 Light gate

A light gate composed of a light source (usually a LED) and a light sensor. The light sensor creates a voltage according to the light intensity upon it.

The lamp and the light sensor of the light gate can be one against the other. The light sensor voltage will be high as long as the lamp light is not blocked. When the light is blocked, the light sensor voltage will drop dramatically.

Another option of the light gate is to place the lamp and the light sensor side by side.

The lamp light will reach the light sensor only when the light gate is near a wall or a hand.

Procedure:

1. Connect the **Lamp**, the **Light sensor** and the **Brain Servomotor** to the base card as in the following picture:

Use the brain servomotor with ID #1.

Important notes:

The base has two 3-wire connectors on its sides. These connectors are for exposed wires included in the kit.

Each wire inlet has two metal plates inside. Pressing the plastic above the inlet with a flat screwdriver, opens these plates.

Press the plastic above the inlet and keep it pressed while inserting the wire to it.

Pay attention to the colors of the wires.

2. Install the ROBO-206 on the base card.

3. Connect the ROBO-206 to the PC using the USB cable.

4. Connect the **Brain Servomotor** to the base terminals as described above.

5. Connect the **Buzzer** to the M3 terminals.

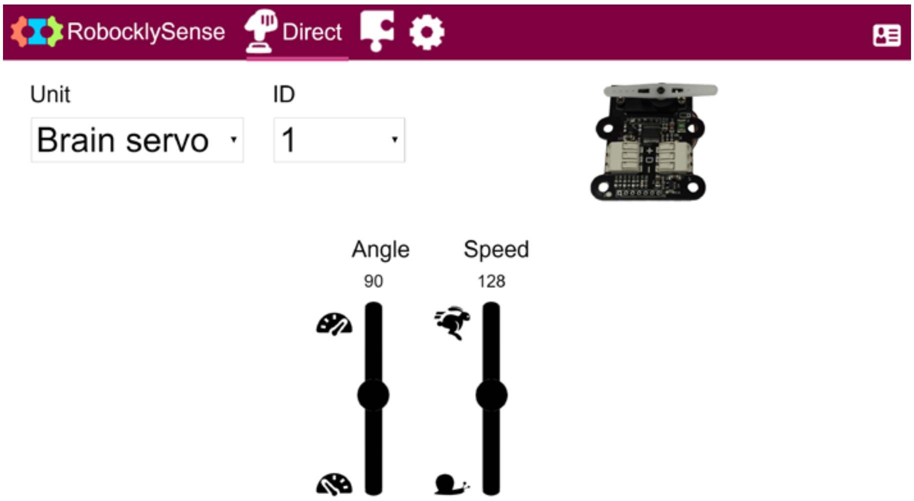
6. Connect the **Lamp** to the M1 terminals.

7. Connect the **Light** sensor to the S1 terminals.

8. Run the RobocklySense software.

9. Test the light gate and the light sensor as described in experiment 1.1.

10. Select **Servo** in the unit field of the right screen.



This screen enables us to set the servomotor angle and the speed of its movement.

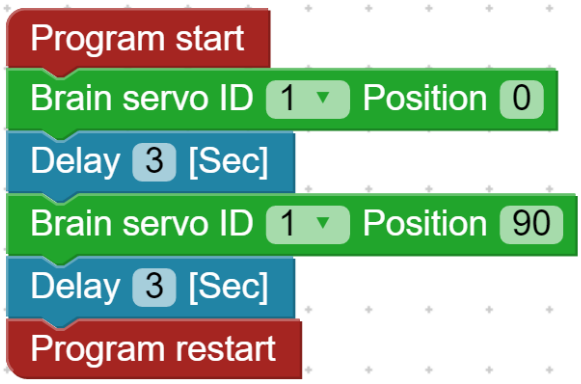
11. Find out the ID number of the Brain Servomotor. The default ID number is 1.

12. Set the ID number of the installed unit.

13. Test the Brain Servomotor by changing the sliders of the angle and the speed.

14. Return to **Block** mode.

15. Build the following program, which moves the servomotor from a 0º angle to a 90º angle in an endless loop.



16. Save the program under the name **SERVO1**.

17. Download and run the program.

    The servomotor shaft moves between the two angles.

18. Stop the program.

1.6.4 Light gate program

1. Move to the **Direct** screen.

2. Turn ON the lamp connected to M1.

3. Read the value of the light sensor connected to S1 and record it.

   We shall call this value the **OFF value**.

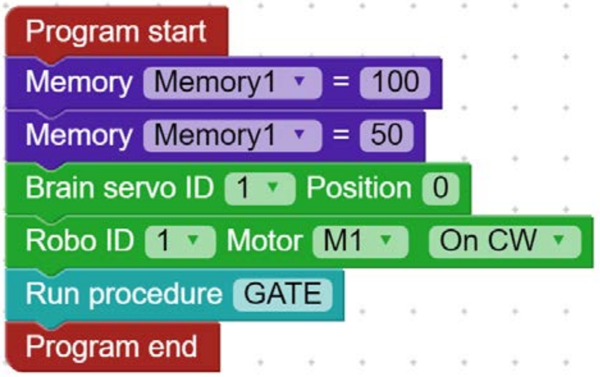
4. Bring your hand close to the light gate and the light sensor.

5. Keep your hand near the light gate and the light sensor, and read the value of the light sensor connected to S1 and record it.

   We shall call this value the **ON value**.

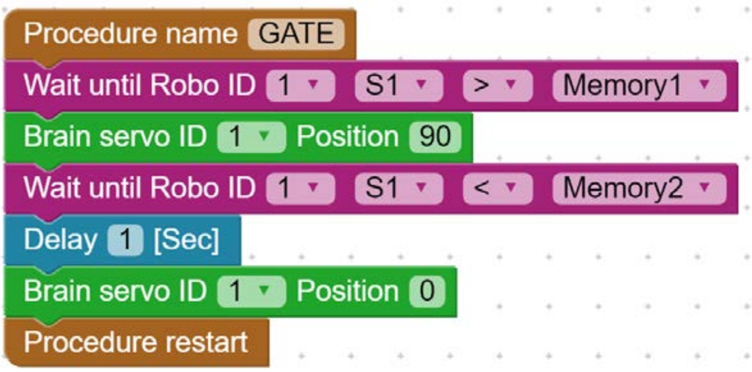
6. Create a main program that does the following:

- Determines memory1 as ON value.
- Determines memory2 as OFF value.
- Moves the servomotor shaft to 0º.
- Turns the light gate ON.
- Calls the GATE procedure.



7. Create the following **GATE** procedure that does the following:

- Waits while the light sensor value is close to the OFF value.
- Moves the servomotor shaft to 90º.
- Waits while the light sensor value is close to the ON value.
- Moves the servomotor shaft to 0º.
- Insert a delay.
- Restart the procedure.



8. Save the program under the name **GATE1**.

9. Download and run the program.

The servomotor lever should move to 0º and the lamp should turn ON.

10. Bring your hand close to the light gate and the light sensor.

    The servomotor shaft should move to 90^o.

11. Take your hand away from the light gate and the light sensor.

    The servomotor shaft should move to 0^o.

12. Repeat steps 10 and 11 and change the Var1 and Var2 setup values until the program works well.

13. Stop the program.

1.6.5 Challenge exercise – Light gate with buzzer

Task 1: Improve the program so before closing the gate, it turns the buzzer ON for 1 second.

Experiment 1.7 – Motorized Cart

Objectives:

- Building a cart that stops in front of a wall.
- Building a cart that moves between two black lines.

Equipment required:

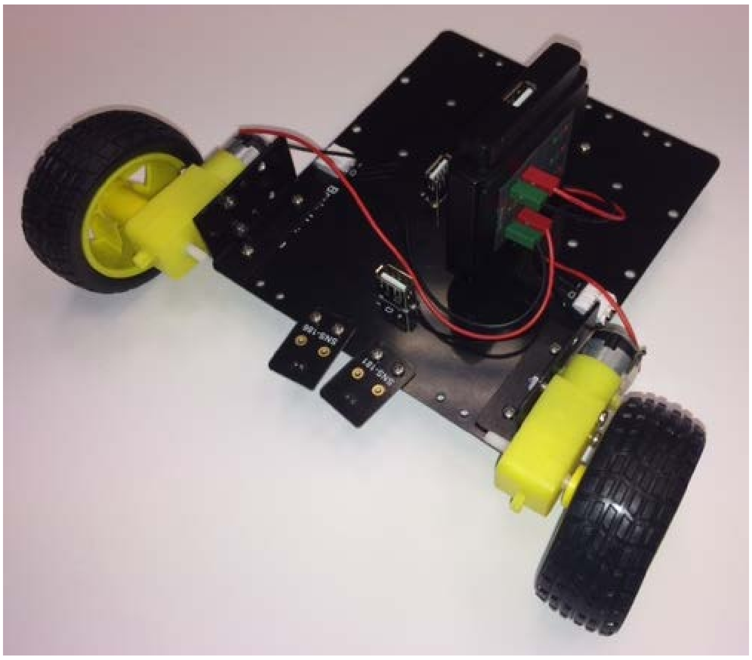
- Computer
- RobocklySense software
- SENSE-MAKE components kit

Discussion:

In this experiment, we will build a cart with optical coupler, which moves between black lines. The positions of the lines limit its motion.

Procedure:

1. Connect the cart components to the base card as in the following picture:



2. Install the ROBO-206 on the base card.

3. Connect the ROBO-206 to the PC using the USB cable.

4. Run the RobocklySense software.

5. Connect the cart components to the ROBO-206 and to the base terminals.

6. Test the components as described in experiment 1.1.

7. Draw two wide lines on an A4 page as follows:

8. Build a program and procedures that move the cart between the two lines.

9. Download, run and test the program.

1.7.1 Challenge exercises – Automatic cart

Task 1: Improve the program so that the cart waits 2 seconds before changing direction.

Task 2: Change the program so moving forward will be slow and backward fast.

Task 3: Change the optical coupler so it will be vertical to the base.

Build a program that moves the cart to a wall, stops when it reaches the wall, goes backward for 4 seconds and goes forward to the wall.

Experiment 1.8 – Movement along a Black Line

Objectives:

- Building a cart that moves along a black line.

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE components kit

Discussion:

In this experiment, we will build a cart with optical coupler, which moves along a black line.

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

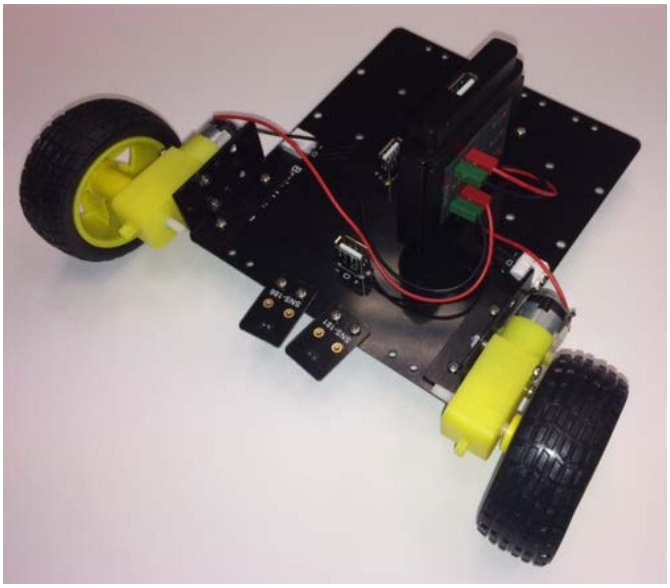
In slow turns, one wheel rotate and the other wheel stops. This way the cart still moves forward while turning.

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

1. Turn left when the cart is on white surface.
2. Turn right when the cart is on black surface.

Procedure:

1. Connect the cart components to the base card as in the following picture:



2. Install the ROBO-206 on the base card.

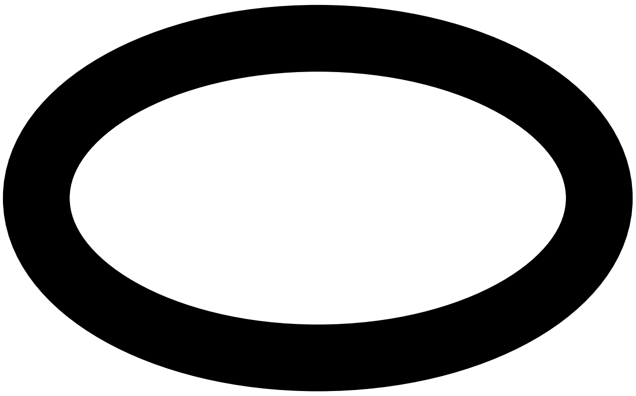
3. Connect the ROBO-206 to the PC using the USB cable.

4. Run the RobocklySense software.

5. Connect the cart components to the ROBO-206 and to the base terminals.

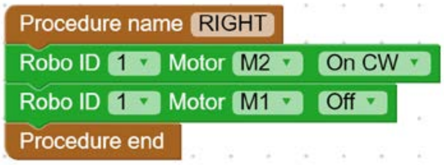
6. Test the components as described in experiment 1.1.

7. Print on a full page a black line as follows:

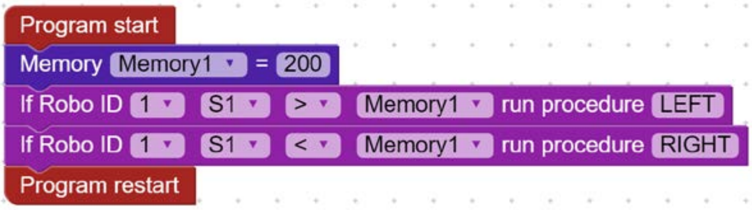


The width of the line should be at least 4cm.

8. Create the **LEFT** and **RIGHT** turn procedures as follows:



9. Create the following main program:



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

11. Put the cart near the black line.

12. Run and check the cart's movement.

13. Change the values in the **If** instructions to create smooth movement of the cart.

14. Change the program so instead of stopping one motor, reduce its speed.

1.8.1 Challenge exercise – Eliminating ambient light effect

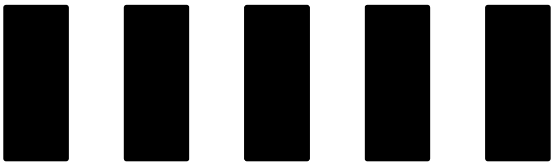
Task 1: Improve the program so the program reads the light sensor twice – one when the lamp is ON and one when the lamp is OFF.

It decides if it is on the black line or not according to the difference between the two readings.

Use memories for the readings and calculations.

Challenge 1.9 – Counting

Draw block lines on a white paper.

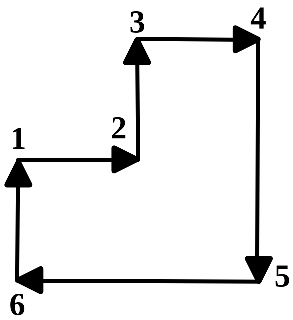


Create a program that moves the cart through the block lines and make it stop on the fourth line.

Use variables to count the lines.

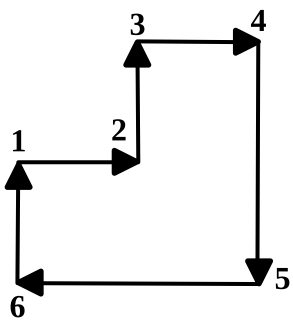
Challenge 1.10 – Automatic movement

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



Challenge 1.11 – Loops

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



Challenge 1.12 – Loops and procedures

Convert each turn and forward movements into a procedure so the main program will have only the loop and run procedure instructions.

The program should do the same as the program in challenge 1.11.

Experiment 1.13 – Automatic Machine

Objectives:

- To construct a machine controlled by a light sensor.
- Control and command of a time dependent system.
- Automatic systems control.

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE components kit

Discussion:

In this experiment, we will construct an automatic machine. This machine simulates washing machine.

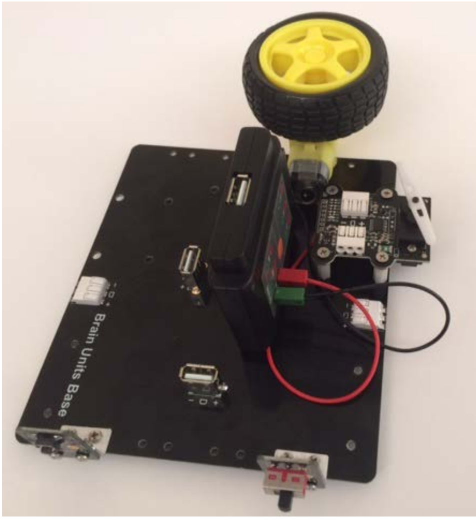
- The brain DC motor simulates the machine motor.
- The brain servomotor simulates the tap with three positions:

  - Clean water (0^o)
  - Tap close (90^o)
  - Dirty water out (180^o)

- The toggle switch simulates ON-OFF switch.
- The tact switch simulates door close.
- The lamp simulates water heater.

Procedure:

1. Connect the machine components to the base card as follows.



2. Install the ROBO-206 on the base card.

3. Connect the ROBO-206 to the PC using the USB cable.

4. Run the RobocklySense software.

5. Connect the washing machine components to the ROBO-206 and to the base terminals.

6. Test the components as described in experiments 1.1 and 1.6.

7. Build program and procedures that does the following:

    - Closes the tap.
    - Waits for the ON switch.
    - Checks that the door is closed.
    - Opens the tap for clean water in for 15 seconds.
    - Closes the tap.
    - Turns ON the water heater for 10 seconds.
    - Turns OFF the water heater.
    - Turn ON the motor CW and CCW 8 times in low speed for 5 seconds for each direction.
    - Opens the tap for dirty water Out.
    - Turns ON the motor CW in high speed for 10 seconds.
    - Stops the motor.
    - Closes the tap.

8. Download, run and test the program.

1.13.1 Challenge exercise – Washing machine

Task 1: Improve the program so it stops the motor when the main switch is OFF.

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 motion sensor (NUL-213)
- NeuLog magnetic sensor (NUL-214)
- Brain tracking unit (SNS-101)

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.

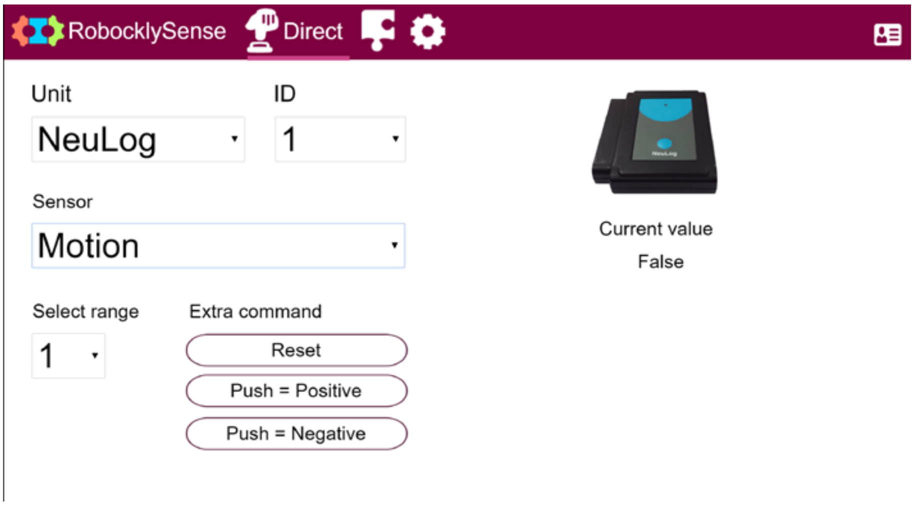
The SENSE-MAKE base has three sockets for NeuLog sensors.

2.3 Changing Brain unit ID

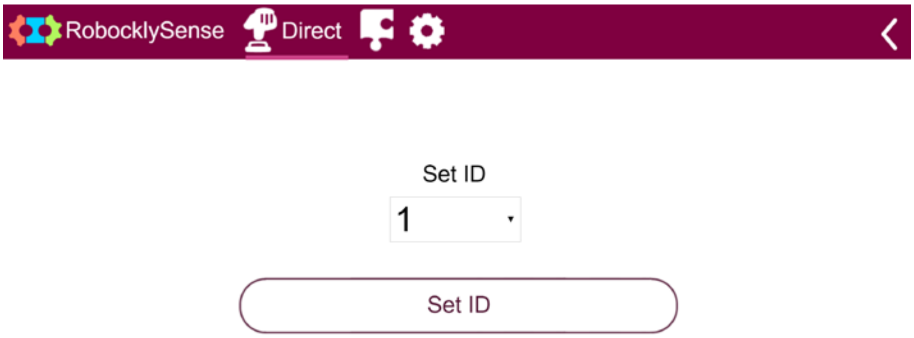
As said before, every brain unit has ID number. The ID number enables us to up to nine brain units of the same kind. We just have to take care that each one of them will have different ID number.

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 – Motion Sensor

Objectives:

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

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE component kit
- BAT-202 NeuLog battery module
- NUL-213 NeuLog motion sensor

Discussion:

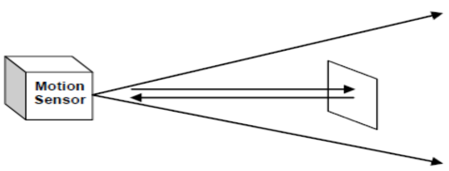
The motion sensor uses an ultrasonic transducer to both transmit an ultrasonic wave, and to measure its echo return. Objects in the range of 0.15 to 10 meters can accurately be measured to give distance, velocity, and acceleration readings using this method.

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).

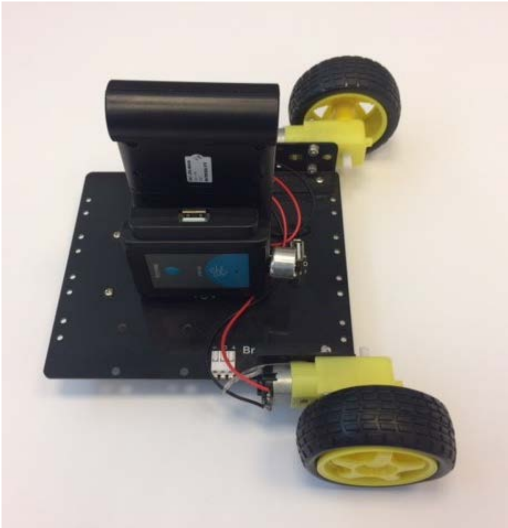
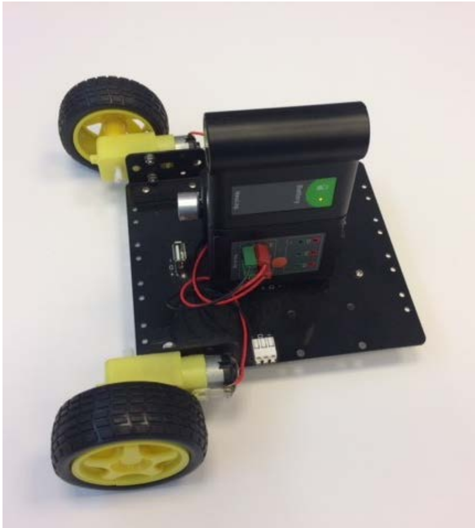
Acceleration is calculated the difference between two successive velocities divided by the time between the samples (according to the sampling rate).

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. Connect the cart components to the base card as in the following picture:

2. Install the ROBO-206 on the base card.

3. Plug the NeuLog motion sensor into one of the base sockets with its transducer directly to the front of the base (as in the picture).

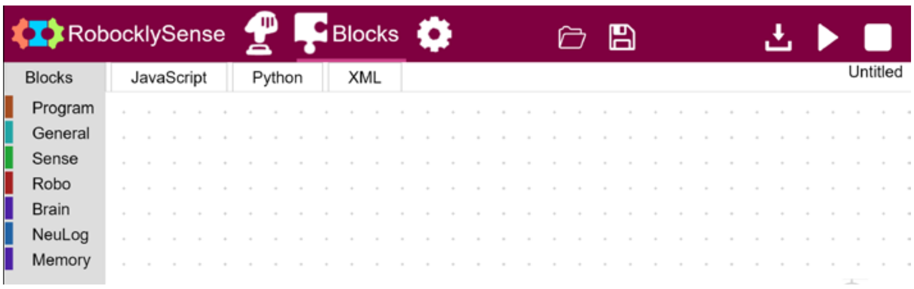
4. Connect the ROBO-206 to the PC using the USB cable.

5. Run the RobocklySense software.

6. Connect the cart components to the ROBO-206 and to the base terminals.

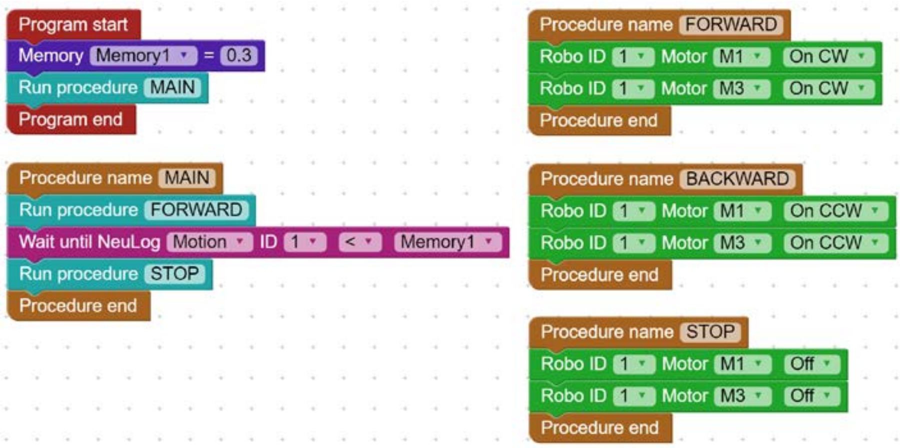
7. Test the components as described in experiment 1.1.

8. Move to **Block** mode.



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

9. Build the following program and its main procedure:



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

11. Place the cart in front of a wall and run the program.

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

2.1.1 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 cart 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 2.2 – Brain Tracking Unit

Objectives:

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

Equipment required:

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

Discussion:

2.2.1 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 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.

2.2.2 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.

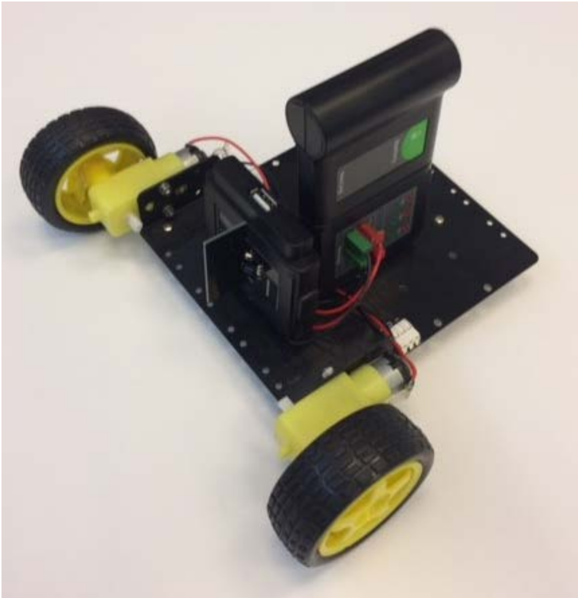
The third IR sensor measures the environment IR light. The brain unit controller uses this measurement to eliminate the environment light.

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
Right (01) – IR transmitter light on the right
Left (10) – IR transmitter light on the left
Front (11) – IR transmitter light at front

Procedure:

1. Connect the cart components to the base card as in the following picture:



2. Install the ROBO-206 on the base card.

3. Plug the Brain tracking unit into the front base socket (as in the picture).

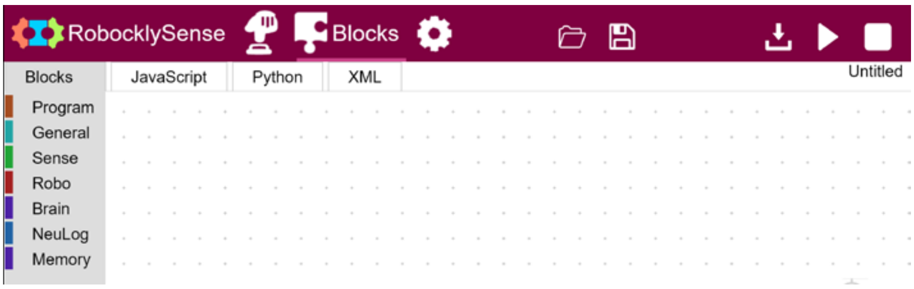
4. Connect the ROBO-206 to the PC using the USB cable.

5. Run the RobocklySense software.

6. Connect the cart components to the ROBO-206 and to the base terminals.

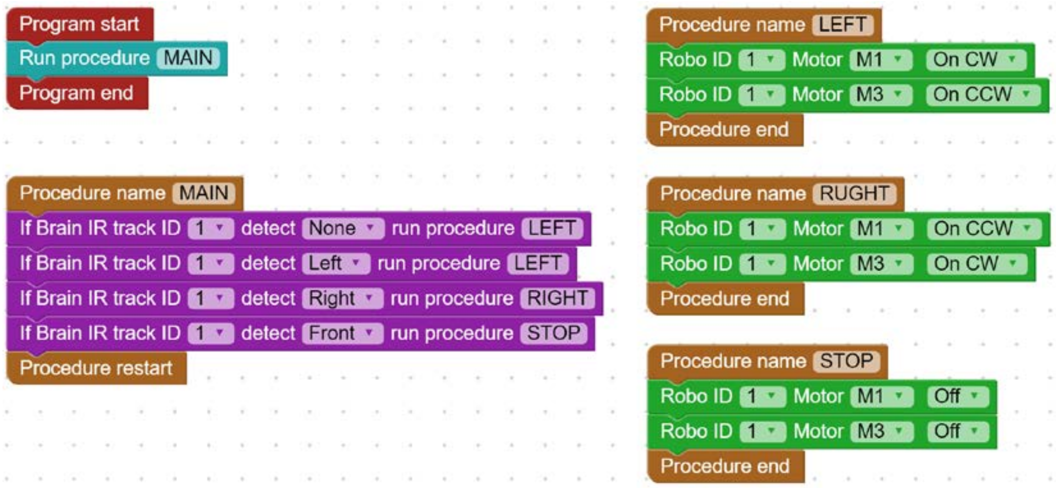
7. Test the components as described in experiment 1.1.

8. Move to **Block** mode.



Create a program that rotates the cart to the left until it "sees" the IR transmitter and then tracks it without moving (just rotates).

9. Build the following program and its main procedure:



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

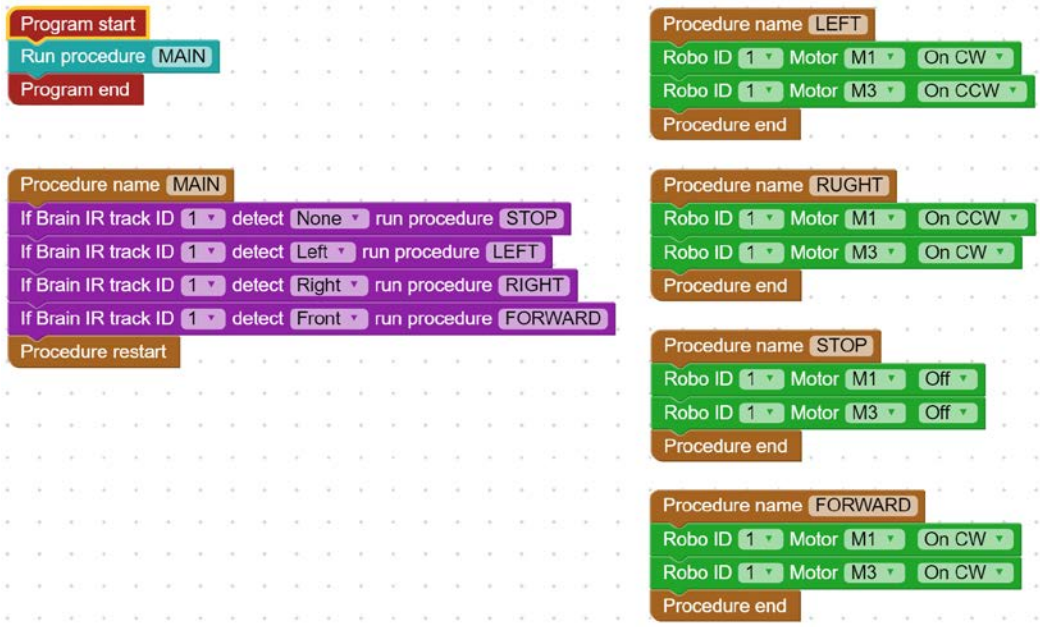
11. Download the program.

12. Place the cart on the floor and run the program.

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

13. Plug the IR transmitter into battery module, move it slowly and check that the cart tracks it.

14. Change the program to the following:



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

2.2.3 Challenge exercise – Tracking a cart with IR transmitter

Task 1: Improve the above program and procedures so the cart 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 2.3 – The Cart and Science Experiment

Objectives:

- The NeuLog light sensor
- Running an experiment while moving
- Using the cart as a USB module with NeuLog software

Equipment required:

- Computer
- RobocklySense software
- SENSE-MAKE component kit
- NUL-204 NeuLog light sensor
- NUL-213 NeuLog motion sensor
- A flashlight

Discussion:

2.3.1 The NeuLog light sensor (NUL-204)

The Light sensor 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.

The light sensor 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.

2.3.2 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.

2.3.3 ROBO-206 as USB module

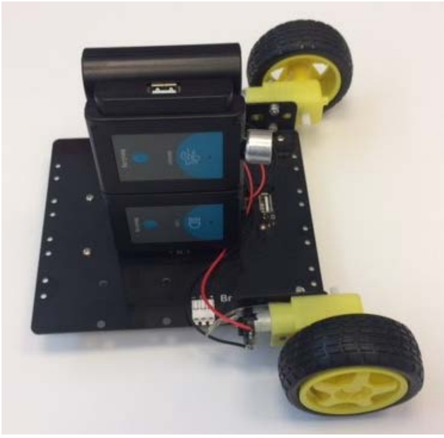
ROBO-206 can act as a USB module for the NeuLog application software.

With this software we can view graphically the experiment results after running the RobocklySense program.

We can, of course, save and use all NeuLog software functions.

Procedure:

1. Connect the cart components to the base card as in the following picture:



2. Install the ROBO-206 on the base card.

3. Plug the NeuLog light sensor and the motion sensor into one of the base sockets (as in the picture).

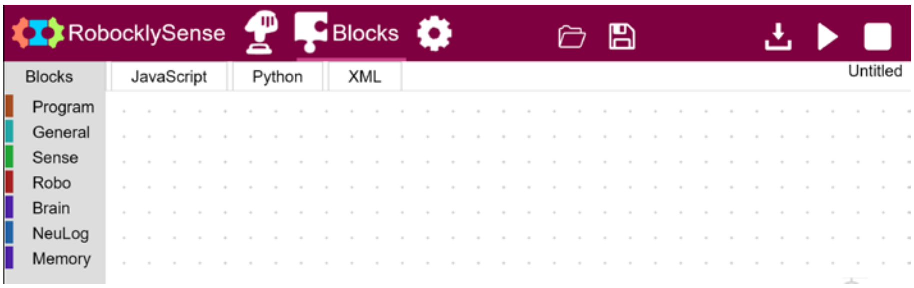
4. Connect the ROBO-206 to the PC using the USB cable.

5. Run the RobocklySense software.

6. Connect the cart components to the ROBO-206 and to the base terminals.

7. Test the components as described in experiment 1.1.

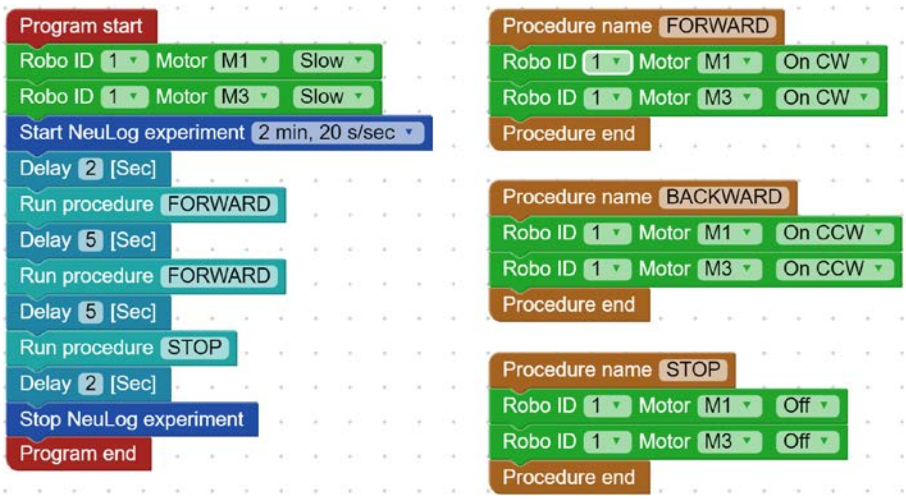
8. Move to **Block** mode.



9. Plug the battery to the second socket.

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

11. Build the following program:



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

13. Download the program.

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

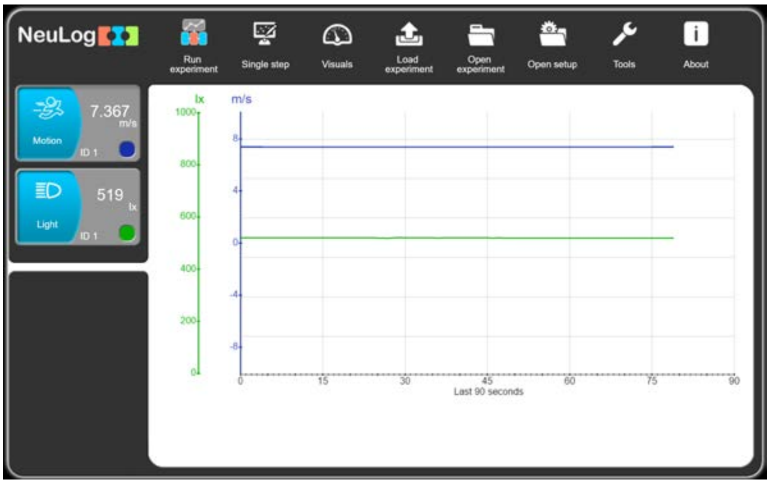
15. Run the program.

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

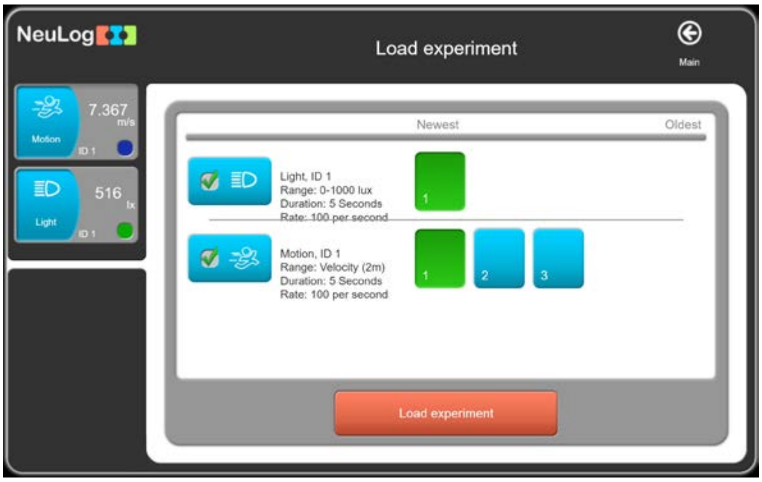
16. Connect the ROBO-206 to the PC.

17. Exit the RobocklySense program.

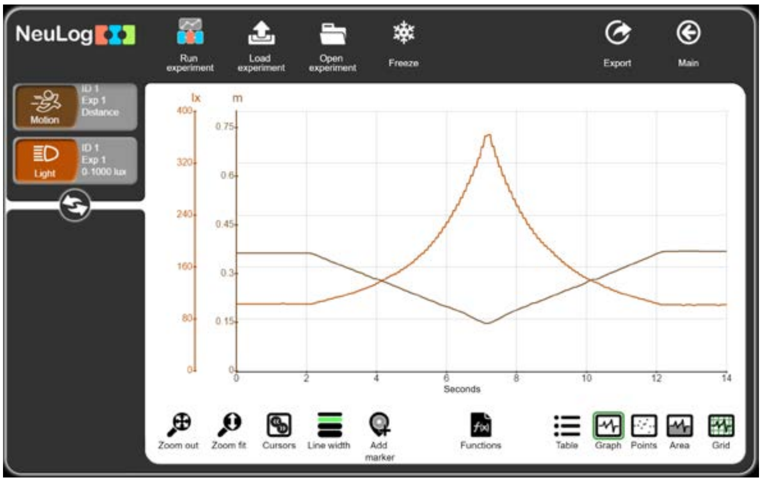
18. Run the NeuLog software and wait for the following screen.



19. Click on the Load experiment button for the following screen.



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



21. 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.

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

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

2.3.3 Challenge exercise – Magnetic fields vs distance

Task 1: Repeat the above experiment with distance sensor and magnetic fields sensor against a wall and a strong magnet.