MOWAY
BEGINNERS MANUAL

Index

1. Prologue

The dawning of a new era; the era of the minirobots. Increasingly more mobile robotics applications enter our daily life. We can currently find robots which help us with simple tasks like cleaning household floors, mowing the lawn or keeping the swimming pool clean. As technology keeps improving, these small devices which blend mechanics, electronics and software are performing more and more complex tasks. They are slowly introducing themselves into our lives in a useful manner and reducing the burden of unpleasant jobs.

It's not too far-fetched to think that the revolution which took place in the IT or telecommunications fields will be repeated with robotics in the next decade. Enough technology is currently available to manufacture these devices and society is also ready to receive them in the market. Yet, a specific catalyst is needed to start this revolution. People also need to be ready and prepared to identify in what fields microrobotics may have an opportunity and which new applications may be interesting to implement.

Up till now processors weren't able to move. But today things have changed. Software is one of the fundamental elements in the world of mobile robotics. The main difference between developing a program for these robots and running it with a personal computer is interaction with the environment. The environment isn't changing randomly in PC applications, so decision making and programming are simplified. On the other hand, when running commands for a minirobot application usually the result is unknown, therefore algorithms have to consider situations with a wider range of possibilities, some of them unexpected.

The mOway robots are tools specifically designed for teaching and research. Their purpose is to bring the world of autonomous robots closer to the teaching centers.

mOway's main purpose is to be a useful tool for those who are being introduced for the first time to the world of the minirobots as well as for those who are already experienced and wish to perform complex collaborative robotic applications.

mOway aims to stimulate enthusiasm for this new and exciting branch of engineering in a prompt and enjoyable way through the practical exercises included in this manual.

This manual has been implemented to assist learning how to use mOway. It provides some basic notions on using mOway and its functions in a quick and clear manner.

This manual is divided in two parts. The first part includes a description of the elements which form part of the robot and their functioning. The second part of the manual includes a series of practical exercises that can be executed with mOway.

2. What is mOway?

mOway is an autonomous programmable small robot designed mainly to perform practical minirobotics applications.

It provides a perfect hardware platform for those wishing to take their first steps within the world of mobile robots as well as for those who have already worked with minirobots and want to develop more complex applications.

The mOway robot is equipped with a series of sensors which aid it to move in a real environment. It also includes a drive unit which moves over smooth terrain commanded by a I2C communications bus. All these peripherals are connected to a microcontroller responsible for governing the robot.

This small robot incorporates I2C/SPI expansion bus options. As an example, a wireless communications module, a video camera or a prototype card can be connected to it as well as any other device considered interesting to perform a certain task.

mOway's external design is very compact, intended to move with grace and style avoiding standstills due to obstacles or corners. This small mobile device has been fittingly called a "pocket robot".

mOway is a perfect tool for those who want to both learn and teach minirobotics. The user will be pleasantly surprised by the speed in achieving results even if this is the first time he/she comes into contact with mobile robots.

3. Robot mOway

This chapter describes each of the parts that constitute the mOway. It is important to highlight that it is not necessary to know the total functioning of the robot to be able to program it, at least not at the level of detail explained here.

The following elements are to be found inside mOway:

- Processor
- Drive system
- Sensors and indicators group
- Power supply system
- An expansion connector

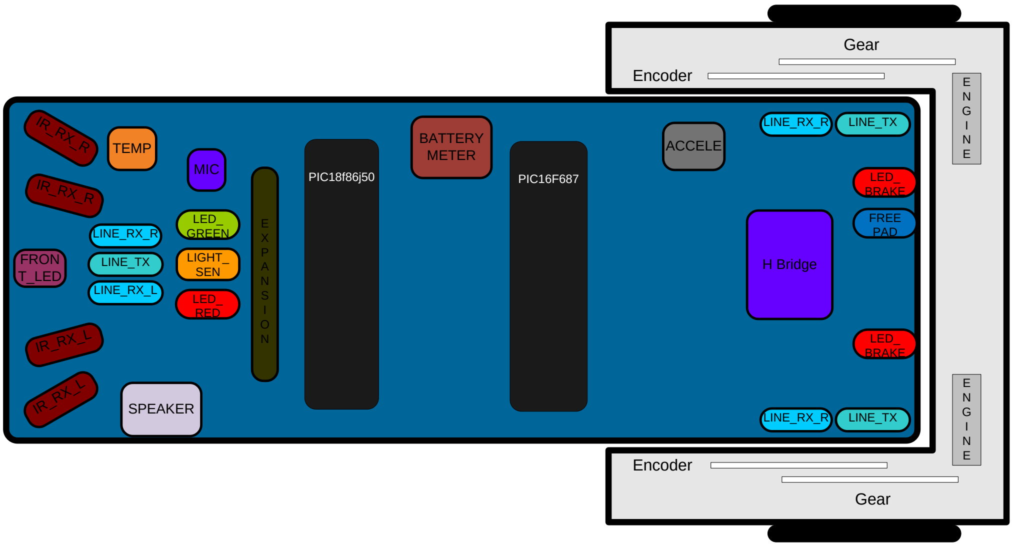


Image 1. Diagram of mOway's parts

3.1. *Processor*

mOways are governed by a 4 Mhz PIC18F86J50 (2.0.1 version) or PIC18F87J50 (2.0.2 version) microcontrollers manufactured by Microchip Technologies. All of the peripherals distributed throughout the whole robot are connected to its input/output ports. Some of them need a digital input or output, others need an analog input or output and others, instead, are controlled by one of the I2C/SPI communication buses. The table below describes how the microcontroller pins are distributed.

3.2. Drive system

To be able to move the mOway uses a double servo-motor group. It includes both an electronic part and a mechanical one. The electronic part is mainly in charge of controlling the motor's speed and the mechanical part allow the mOway to move unhindered over different terrains with adequate power.

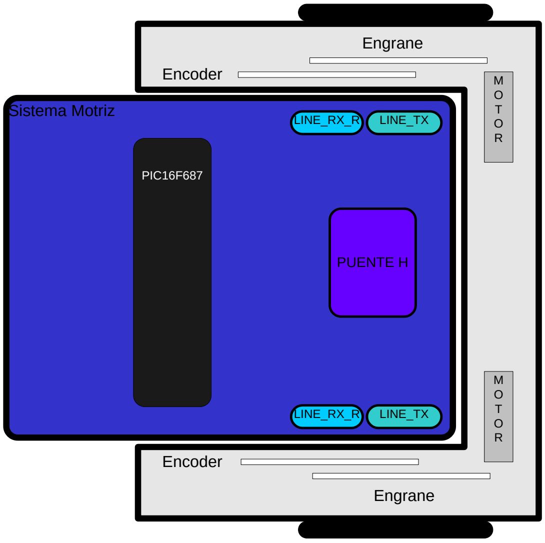


Image 2. Drive system: electronic and mechanical

The servo-motor group includes different features:

1. **Speed control**: controls the speed of each motor.
2. **Time control**: controls the time for each command with a 100 ms precision.
3. **Travelled distance control**: Controls the distance travelled by each command with a precision of 1 mm aprox.
4. **General speedometer**: counts distances travelled since the initial command.
5. **Angle control**: controls the angle when the mOway rotates.

The microcontroller sends the I2C command to the drive system that controls the motors and therefore releasing the main microcontroller so it can carry out other tasks.

Speed control is carried out by means of proportional control with negative feedback from the encoders' signal. The illustration displays the controlling system. The microcontroller feeds the motors through an H bridge controlled by pulse width modulation (PWM) signals. Wheel rotation is monitored by an encoding sticker and an infrared sensor. When the sticker shows its black segment, the logical output shall be 1 and when it shows the white sector the output shall be 0. The microcontroller analyzes these signals (it can determine the exact wheel speed by measuring the pulse width) and acts on the motors. This way, the mOway will be able to keep the speed constant on any surface.

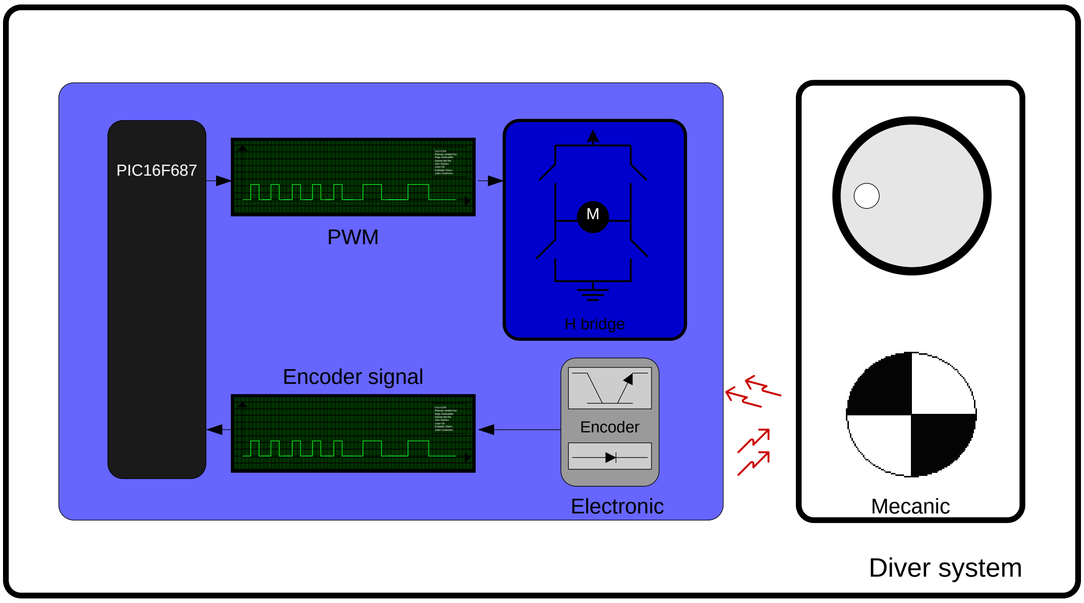


Image 3. Motor control

To send a movement command to the robot, via the main microcontroller, all we need to do is send the movement command parameters. To this end some libraries were designed in assembly and C language to simplify communications through some functions which are responsible for I2C communications. The format for these frames is explained in the motors and drive system library section.

The table below describes connections between the main PCB and the servo-motor unit.

3.3. Sensor and indicators group

This group consists of different luminous sensors and indicators, connected to the mOway microprocessor, through which the robot interacts with the external world:

- Two line tracking sensors.
- Four obstacle detection sensors.
- A light sensor.
- An expansion connector.
- Four LED diodes.
- Temperature sensor.
- Speaker.
- Microphone.

- Accelerometer.
- Battery level.

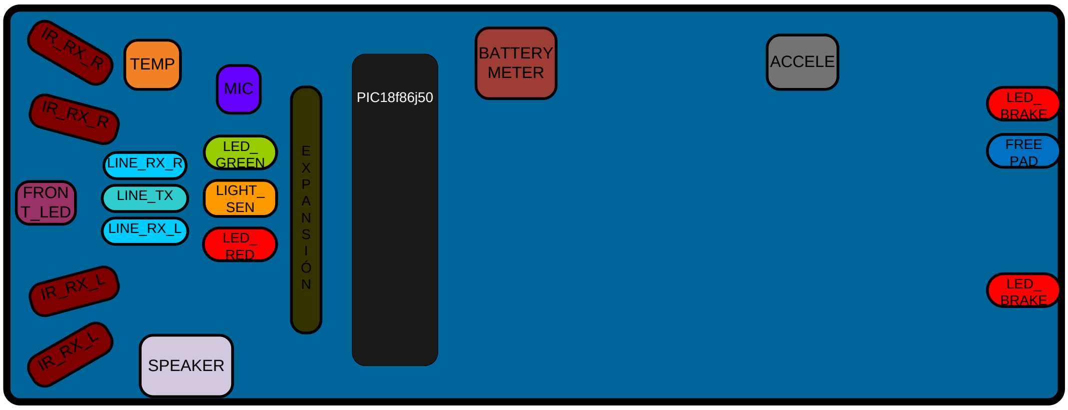


Image 4. Sensors and indicators group

3.3.1. Line sensors

The line tracking sensors are two reflection optocouplers mounted on the top front part of the robot. They use infrared light reflection to detect the colour of the terrain at the point where the robot is.

These two sensors are connected to two microcontroller analog ports so strong terrain contrasts, like white lines on black backgrounds, can be detected. They are also capable of distinguishing different tones.

The Vishay CNY70 sensor has a compact construction where the emitting light source and the detector are arranged in the same direction to be able to detect by using the reflective IR beam the light reflected in the terrain.

In the images below the three possible cases can be observed:

1. **Clear surface**: A white surface reflects all the infrared light and therefore we obtain a low voltage reading at the transistor’s output when in regular mode.

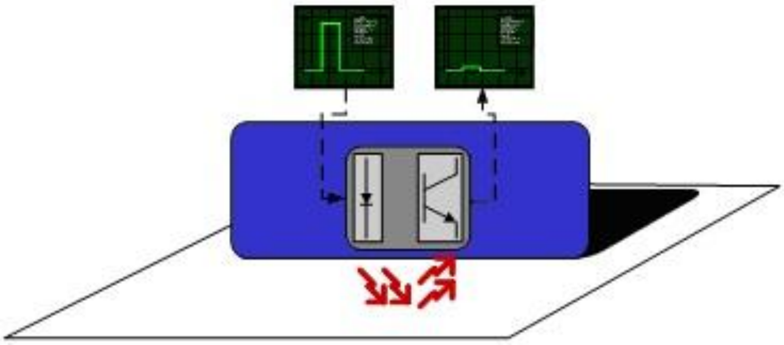


Image 5. Line tracking sensor on a clear surface.

- **Colour surface**: A colour surface reflects part of the emitted light obtaining an intermediate voltage at the microcontroller's analog channel input. This way colours are easily identified[1].

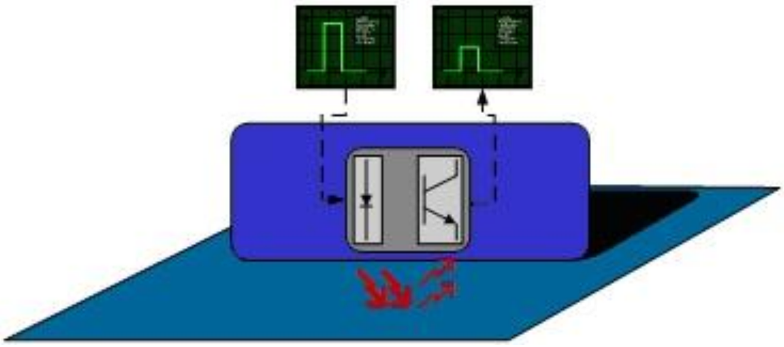


Image 6. Line tracking sensor on a colour surface.

1. **Dark surface**: A dark surface reflects very little light obtaining a high voltage reading at the sensor's output.

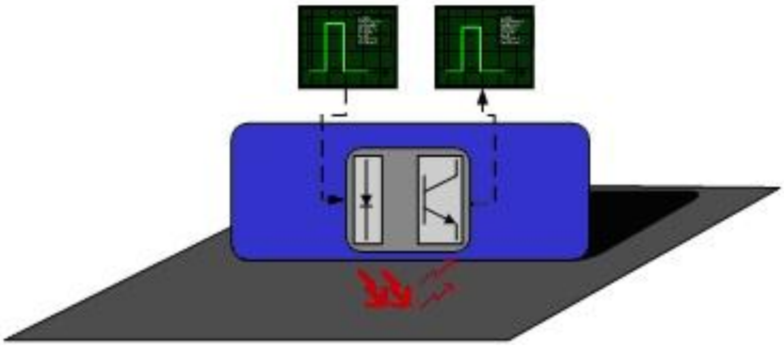


Image 7. Line tracking sensor on a dark surface.

[1] Due to CNY70 tolerance two different sensor can differ.

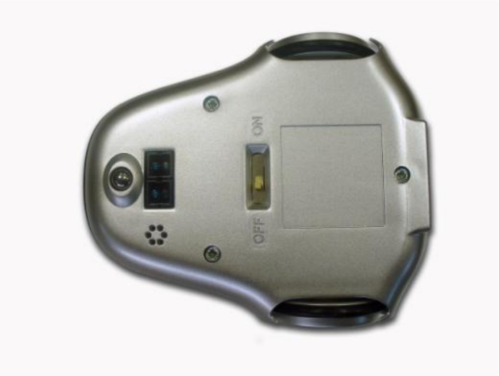


Image 8. Location of line sensors

3.3.2. Obstacle detection sensors

Similar to line tracking sensors, obstacle detection sensors also use infrared light to detect objects located in front of the mOway. The sensor includes two infrared light-emitting source (Kingbright KPA3010-F3C) and four receivers placed on both sides of mOway.

The output of the Sharp PT100F0MP receivers are connected to the microcontroller's analog inputs so it can detect the presence of any object (digital mode) and also measure how far away it is (analog mode)[2].

The sensor functions similarly to the line tracking sensor. The light emitter generates a 70us pulse which makes the receiver to capture any obstacle using a filtering and amplifying stage. Once the signal is processed electronically, the PIC can measure it by means of the ADC or as a digital input. The digital distance range is close to 3cm and a bright environment is recommended to enhance infrared light reflection.

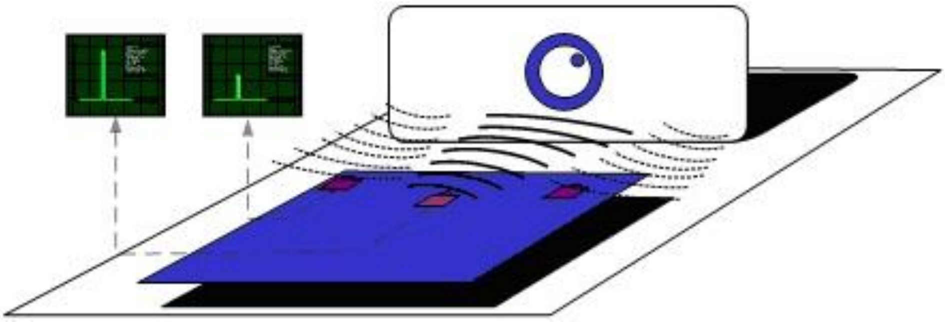


Image 9. Obstacle detection sensor

[2] Due to tolerance two different sensors can differ from each other.

Image 10. Location of Obstacle Sensor

3.3.3. Light sensor

This sensor makes mOway to recognize the light intensity that enters through a small half moon-shaped opening on the top part of the chassis. Since it is facing forward it enables it to detect where the light source is located and to act accordingly.

The output of the AVAGO APDS-9002 sensor is connected to the analog port of the microcontroller so that with a simple reading of the ADC we can register the light intensity level and any change in intensity levels based on the last reading[3].

Image 11. Location of Light Sensor

3.3.4. Expansion connector

This connector allows the mOway to connect with any commercial modules or electronic circuits the user may choose.

As shown in the above table, it is possible to connect commercial SPI devices. On the other hand, the RF BZI-RF2GH4 module available in the market is totally compatible with mOway and with specific libraries. This module enables the mOway to communicate with other robots and with a PC via the RFUSB. With this module it is possible to create complex collaboration applications without having to worry about complicated wireless communications.

[3] Top two-color LED has to be switched off to have a valid measure.

Image 12. RF modules into expansion connector.

3.3.5. Temperature sensor

mOway has installed as a temperature measurer an NTC thermistor from Murata, a semiconductor whose electrical variable resistance decreases as temperature increases. The sensor is located in the front part of the robot, very close to obstacle sensor.

The thermistor is connected to the analog port of the microcontroller so that with a simple reading of the ADC it is possible to get the temperature value in that moment and notice any change in it since the last reading[4].

3.3.6. Speaker

The CMT-1102 speaker from CUI INC directly connected to the microcontroller, is capable to play tones from 250 Hz to 65 KHz.

3.3.7. Microphone

The CMC-5042PF-AC microphone from CUI INC enables the robot to detect sounds from 100 Hz to 20 KHz.

The output is directly connect to an analog input of the microcontroller so that it is capable to detect not only if there is sound or not (digital mode) but also the intensity of the sound with a simple reading of the ADC (analog mode).

[4] Temperature measured by the sensor can be 5ºC higher than external temperature.

3.3.8. Accelerometer

An accelerometer is a device that measures acceleration and the gravity induced forces: the movement and rotation. There are many types of accelerometers, most of them based on piezoelectric crystals, but their size is too big. Because of that, it was tried to design a small device in the field of microelectronics, which might improve the applicability. Then, the MEMS (Microelectromechanical Systems) accelerometers were created.

An easy way to create an accelerometer is measuring changes in a capacitor. Capacitors can work as sensors or as actuators. In the case of mOway, it is a capacitive accelerometer, which consists of two capacitors displaced in differential mode whose electrical capacity changes as the acceleration varies.

By measuring X, Y, Z axes of the MMA7455L accelerometer from FREESCALE Semiconductor, it is possible to know if mOway is correctly positioned, inverted or tilted.

3.3.9. Battery level

The robot has a LiPo cell battery rechargeable. For proper operation of the microcontroller, the battery is connected to one of its analog inputs through a splitter. Thus, with a reading of the ADC battery level can be measured.

3.3.10. Front LED

The front LED is a white LED placed on the front side of mOway. The output of the OSRAM LW A6SG LED is connected to a digital output of the microcontroller.

3.3.11. Top two-colour LED

This double indicator and the light sensor share the same opening on the top part of the robot. They are connected to two microcontroller digital outputs[5].

[5] Please note that since they share the same opening as the light sensor it is fundamental to switch them off when wanting to perform a light intensity reading.

Image 13. Robot with Front LED and red LED switched on

3.3.12. Brake LED

The brake LED is double indicator placed on the back side of mOway. The output is connected to one digital outputs of the microcontroller.

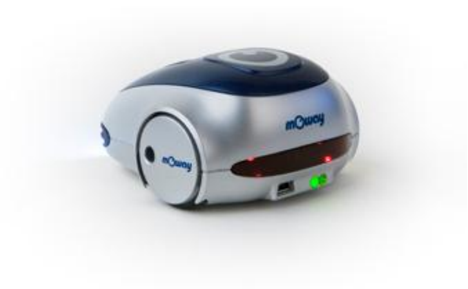


Image 14. Brake LED location. Switch on green LED.

3.3.13. Free Pad

mOway has implemented a free Pad to allow expert users to connect their electronics. It is accessible opening the robot and it´s located near brake LED[6].

3.4. Power Supply System

mOway's battery is located inside and accessible only by disassembling the product. It is a small rechargeable LiPo cell.

The battery can be charged via a computer's USB port through the mOway's MINI-USB-B port. There is no need to wait for the battery to be completely discharged,

[6] Advanced users only

as it can be plugged in any time since these batteries do not have memory effect (also known as lazy battery effect). These batteries are a perfect power source for mOway due to their small size, lightness and flexibility.

Battery duration depends to a great extent on the active sensors and the amount of time the motors are used. Charging lasts about 2h.

Power supply system controls two LED located in the back part of the robot[7]. Green LED indicates that mOway is switched on and red LED indicates that the battery is charging. When the battery is full red LED will switch off[8].

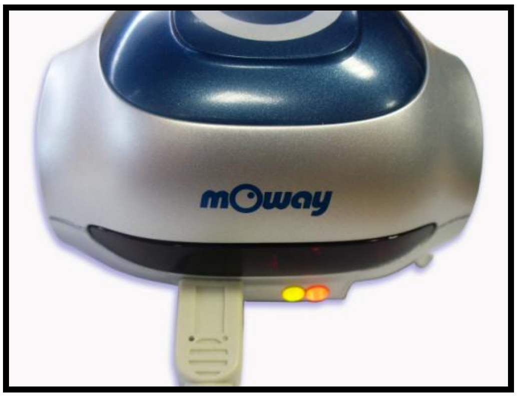


Image 15. Charging (red) and switched on (green)

3.5. *RF module and RFUSB*

RF module[9] establishes communication between other mOways or with PC using RFUSB[10].

[7] These LEDs can´t be controlled by the user.

[8] This LED can swap between on and off when the battery is fully charge because there is energy consumption when mOway is plugged.

[9] Available in some packs

[10] Available in some packs

Image 16. RF module

RF module is connected in expansion connector and it is very easy to use with MowayWorld.

Image 17. RFUSB

The **BZI-RF2GH4** radio-frequency communications module is based on the nRF24L01 transceptor manufactured by "Nordic Semiconductors". This integrated circuit has been fitted with all the logic required to establish wireless bidirectional communications with acknowledgement of receipt. Communications with the microcontroller is made via an SPI bus.

The main characteristics of the BZI-RF2GH4 module are as follows:

- Low consumption.
- Working frequency: 2.4GHz,
- Transmitting power between-18 and 0 dBm,
- Transmission speed between 1 and 2 Mbps,
- 128 in transmission channels selectable by the SPI bus.

In addition to the CI nRF24L01, the **BZI-RF2GH4** is also fitted with all the associated electronics for its correct operation plus a microstrip antenna on the same board with the impedance adaptation network. In this way the user can forget completely about the hardware required to implement the radio part of his application.

As interface, the device has four pins available for the SPI bus, two more pins for controlling the module and another two for the supply.

In order to facilitate the handling of the module, a number of libraries have been developed to simplify and shorten the development time of wireless applications with these modules.

3.6. mOway Camera module

Thanks to the camera module (**mOway Camera Module**[11]) it is possible to display on the computer what mOway is "watching" . Camera board sends images wirelessly to the video receptor **mOway Camera Board**[12].

Image 18. Camera module

Camera Module is connected to expansion connector of mOway robot. It is controlled by robot programming. It has a connector on the back, so that other circuitry can be attached (RF module or user circuitry).

Camera control is performed by Microchip **MCP23S08** device, which is an input/output port controlled by SPI. The basic functions are the following:

- Turn on: When camera is on, the LED of the module is on. The camera sends images through RF to video receptor.
- Turn off: When camera is off, the LED of the module is off. The camera RF transmission ends.
- Channel change: When the camera channel changes, the LED blinks.

As interface, the device has four pins available for the SPI bus, two more pins for controlling the module and another two for the supply.

Other wireless transmitters can affect the images quality. If so, change the channel of transmission (both camera and video receptor).

NOTE: Both camera transmission and mOway RF module transmission are in the same frequency band. So that, when camera is active, mOway RF module reception distance decreases.

[11] Available in some packs
[12] Available in some packs

3.7. mOway Camera Board

mOway Camera Board is the video receptor. It receives the images from the Camera Module through RF and sends them to PC through USB.

Image 19. mOway Camera Board video receptor

It has a channel selector for choosing the RF communication channel with Camera Module (channels 1 to 4). Both Camera Module and video receptor must have selected the same channel.

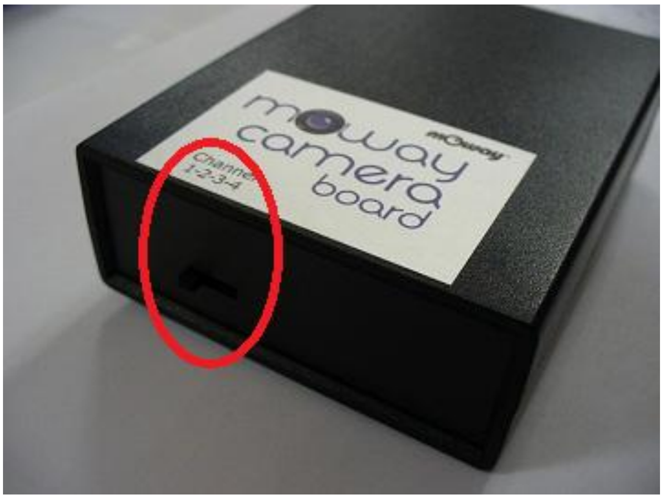


Image 20. Channel selector

Video receptor is connected to a USB port of the PC. First time it is connected, a message for installing drivers can appear. In this case, select the option of installing software automatically.

NOTE: While video receptor is receiving images, the casing heats up. This is a normal functioning.

4. First Steps

4.1. mOway Pack installation

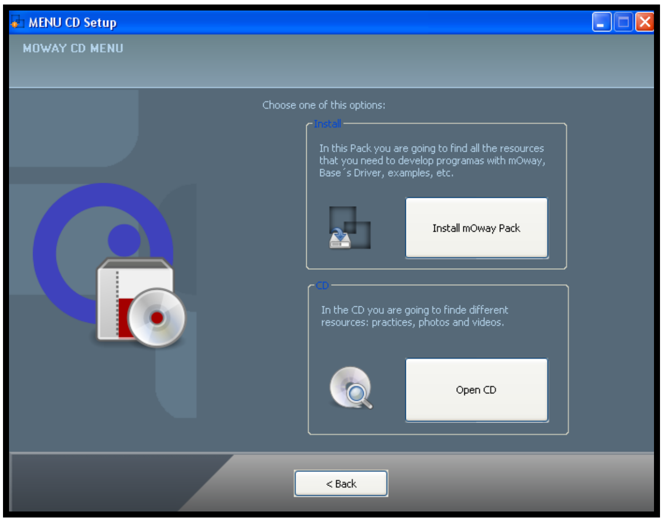
In mOwayPack (available in the webpage www.moway-robot.com or in the installation CD) you will find the software, mOway´s libraries, test programs and documentation. Following setup steps you will have all the resources:

- Beginner’s and User manual.
  - Beginner’s manual includes all you need to start working with mOway.
  - User manual contains detailed description of the robot.
- MowayWorld software.
  - This software controls all aspects of the robot: program download, battery charge control, radio control, RFUSB[13] management and C or assembler programs download.
- Reference projects in assembler, C and MowayWorld.
  - Example projects to start working with mOway easily.
- RFUSB Driver
  - Driver for RFUSB that establishes the communication between robots and PC.
- mOway Camera Board Driver
  - Driver for mOway Camera Board[14] that makes it possible to grab images from mOway camera to display on PC.

If a security warning message appears during installation, please click on “install driver anyway”. MowayWorld software is safe.

[13] Module not available in all kits

[14] Module not available in all kits



Choose one of this options:

In this Pack you are going to find all the resources that you need to develop programas with mOway, Base´s Driver, examples, etc.

In the CD you are going to finde different resources: practices, photos and videos.

Image 21. CD menu

4.2. Download a program to mOway

Program download process is executed in MowayWorld. This application can download to the robot MowayWorld projects, assembler projects (compiled with MPLAB or gputils) and C (C18 compiler) projects.

Steps to download a program to mOway:

- Connect mOway to the PC through USB. The robot doesn´t need any driver.
- Open MowayWorld application.
- Open or create a project in MowayWorld, or import a “.hex” file from assembler or C project.
- Click on “Program mOway” icon. If a “.hex” file has been imported the download progress will start automatically.
- Disconnect the robot and turn it on.

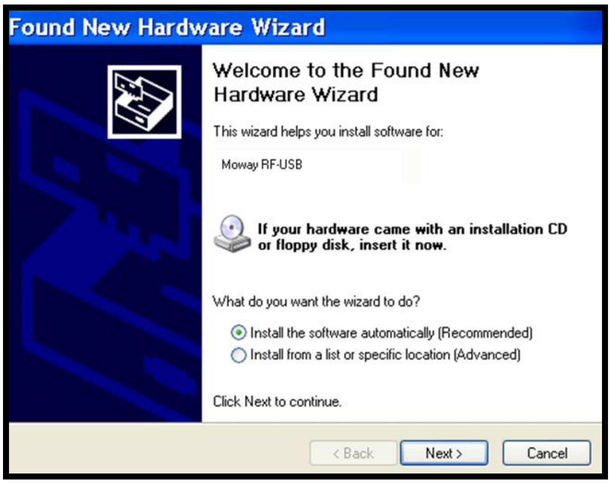
mOwayPack provide several compiled projects to check sensors, drive system, RF communication and camera usage.

4.3. RFUSB installation

This is a device that makes it possible to communicate the PC and mOway. A driver that is included in mOwayPack is required:

- The first time the RFUSB is connected, the PC will detect it as a new device and an “Assistant for new hardware found” message will be displayed. Select the *No, not this time* option.

- In the following window select the recommended option: Install software automatically.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

Moway RF-USB

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

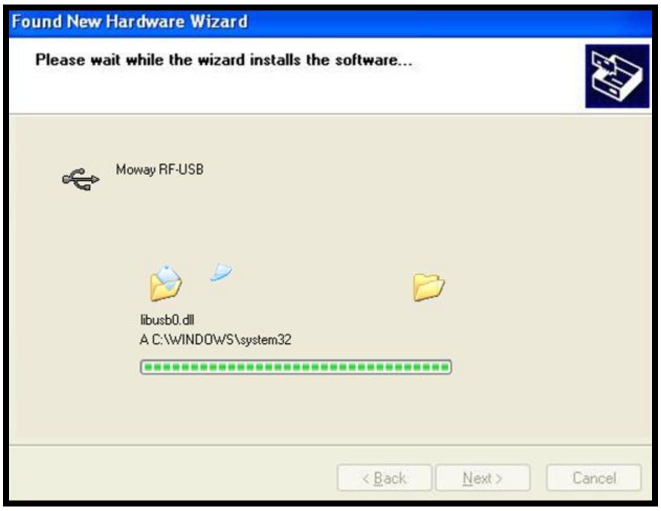
● Install the software automatically (Recommended)

○ Install from a list or specific location (Advanced)

Click Next to continue.

Image 22. Driver installation Wizard

- Now the installation process will begin.



Found New Hardware Wizard

Please wait while the wizard installs the software...

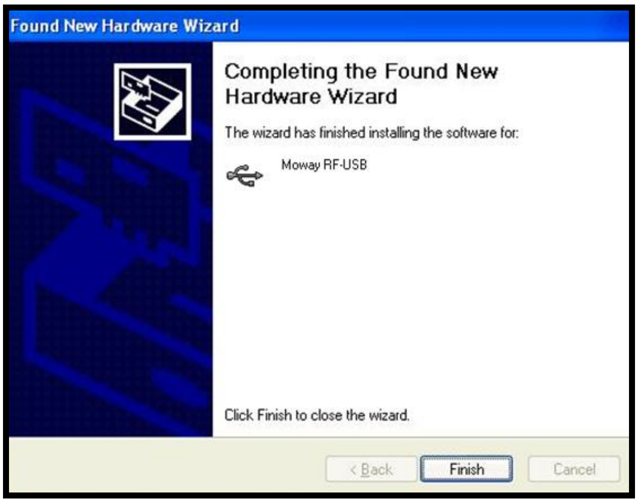
Moway RF-USB

libusb0.dll

A C:\WINDOWS\system32

Image 23. Windows XP driver installation

- Assistant will then indicate that the hardware is installed.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

Moway RF-USB

Click Finish to close the wizard.

Image 24. Driver installed in Windows XP

- Check if mOway’s software has detected the RFUSB.

4.4. mOway Camera Board drivers installation

As with RFUSB device, a driver that it´s included in mOwayPack is required to use video receptor:

- The first time the **mOway Camera Board** is connected, the PC will detect it as a new device. Driver installation runs automatically. If it doesn´t, an “Assistant for new hardware found” message will be displayed. Select the *No, not this time* option.

- In the following window select the recommended option: Install software automatically.

- Now the installation process will begin.

- Assistant will then indicate that the hardware is installed.

- Check if mOway’s software has detected the **mOway Camera Board.**

5. MowayWorld programming

MowayWorld application makes it possible to design programs by means of flowchart diagrams, so that mOway robot can be programmed very easily. Different modules represent functions that control robot's sensors and actuators. These modules are joined by means of arrows to define the program flow. Previous programming knowledge is not essential to use flowchart programming.

5.1. MowayWorld workspace

The following lines describe the different parts of MowayWorld workspace.

5.1.1. Toolbar

Toolbar makes it possible to manage the project, edit the flowchart, create variables, program mOway robot, change MowayWorld language and many other functions.

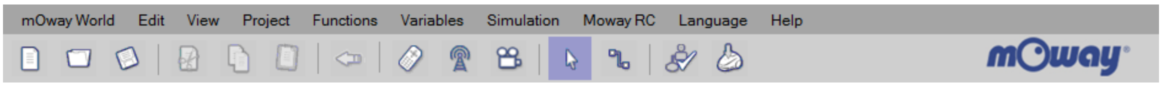


Image 25. Toolbar

5.1.2. Flowchart Editor

The Flowchart Editor window is where modules are placed and connected to develop the program. When a new project is created, this window is empty except for the starting point. In flow diagrams there is always a "starting point" from which the program begins.

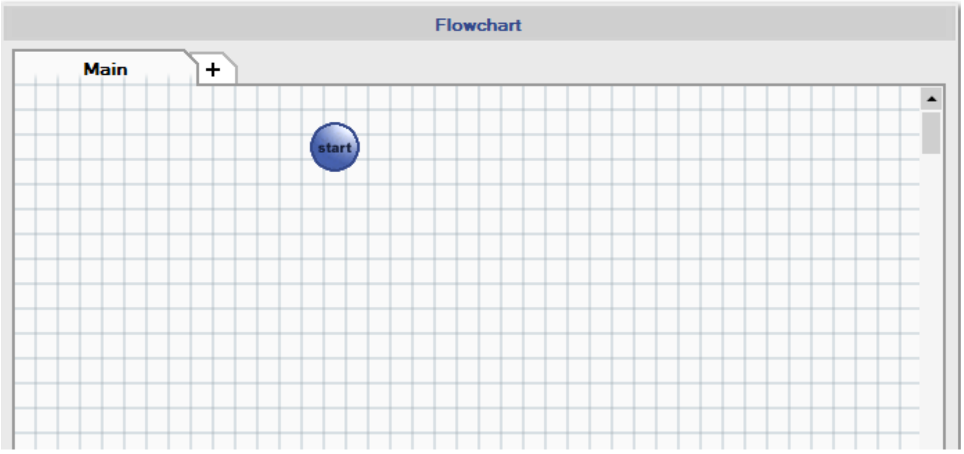


Image 26. Flowchart Editor window

5.1.3. *Tools*

On the left side of the workspace we can find the Tools section. Here there are all the modules to control mOway, such as movement actions, checking sensors, communication, and so on. Modules are grouped by their type of function (Actions, Sensors, Data, Flowchart Control and Expansion connector).

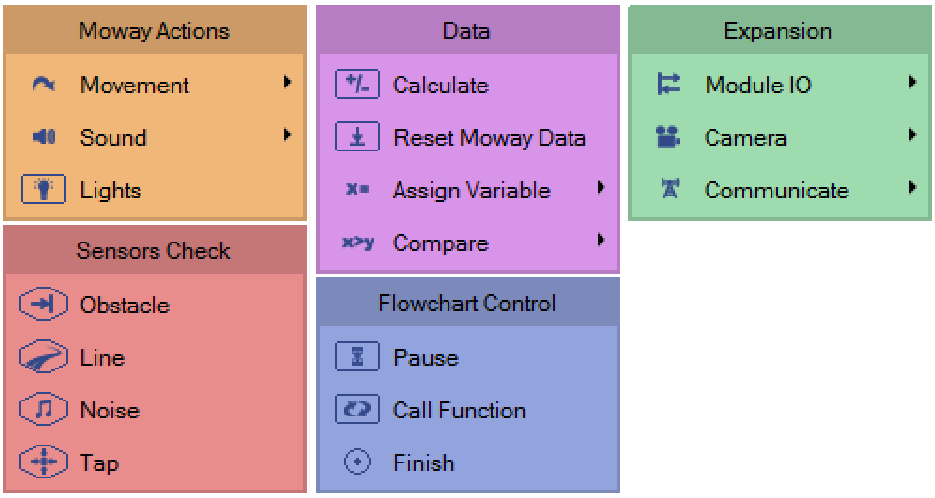
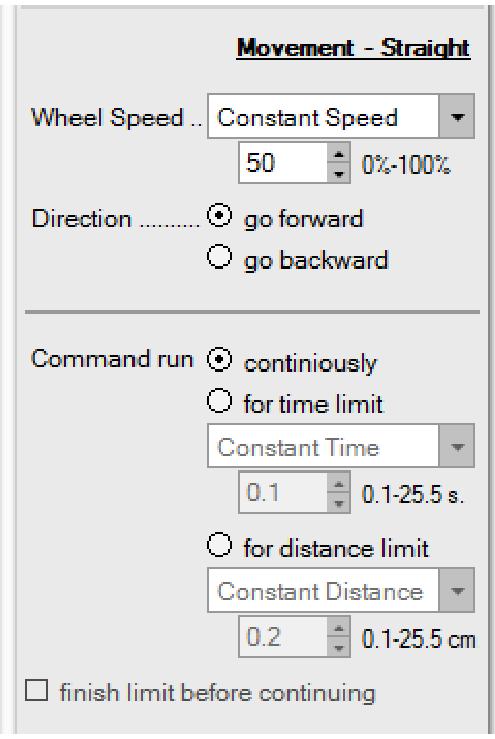


Image 27. Toolbox section

A module is inserted by just dragging and dropping it in the flowchart editor. This module function can be configured by making double click on it, or in the “Properties” window (see next chapter). Some modules are grouped by a more specific type (Movement, Sound,…). This is represented by a little black arrow in the toolbox module types.

5.1.4. *Properties*

When a module in Flowchart Editor is selected, the “Properties” window will appear on the right side of the workspace. This makes it easier and faster to change the modules configuration.



Movement - Straight

Wheel Speed .. Constant Speed

50 0%-100%

Direction .... ● go forward
○ go backward

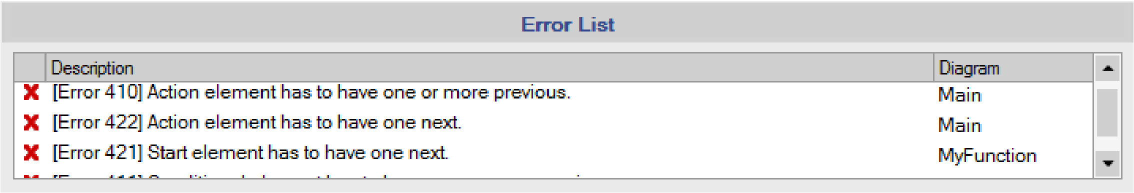
Command run ● continiously
○ for time limit
Constant Time
0.1 0.1-25.5 s.
○ for distance limit
Constant Distance
0.2 0.1-25.5 cm

□ finish limit before continuing

Image 28. “Straight” module properties

5.1.5. *Error List*

Error List window shows the errors found in the diagram. It includes a description of the error (Description) and the diagram in which the error has produced (Diagram). A diagram with errors will not be downloaded to the robot.



Error List

	Description	Diagram
✗	[Error 410] Action element has to have one or more previous.	Main
✗	[Error 422] Action element has to have one next.	Main
✗	[Error 421] Start element has to have one next.	MyFunction

Image 29. “Error List” window

5.1.6. Arrows

Modules execution order is established by means of arrows. In order to connect the modules, following steps must be taken:

- Place the cursor over a module until red and white marks appear.
- Click on one of these marks.
- Click on one of the next module mark.

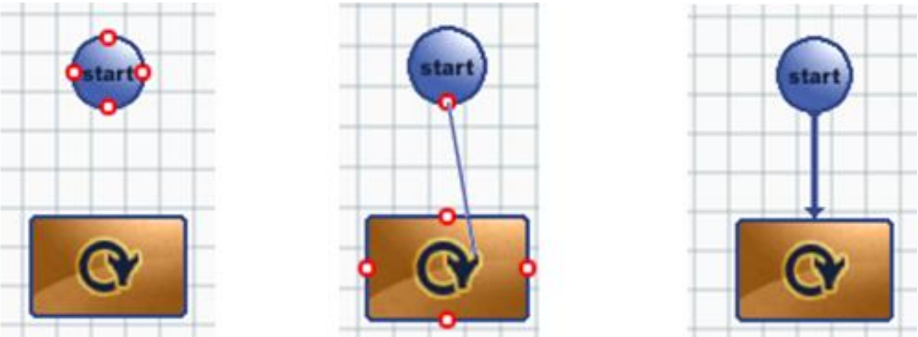


Image 30. Arrow drawing

“Connector” tool makes easier to draw arrows. It is located on Toolbar. For selecting diagram elements it is necessary to change to the default “Cursor” tool.

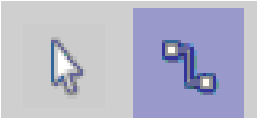


Image 31. Cursor tool and Connector tool icons

All the modules of the diagram must have at least one input arrow. In addition, they must have one output arrow (two output arrows in case it is a conditional module).

The conditional modules (oval shaped) have two outputs, depending on the condition. If the condition is true, the program will flow through “true” path (arrow with a green mark). If the condition is false, the program will flow through “false” path (arrow with a red cross).

In order to describe a loop, the procedure is similar but in this case the arrow starts on a module mark and finishes on another mark of the same block.

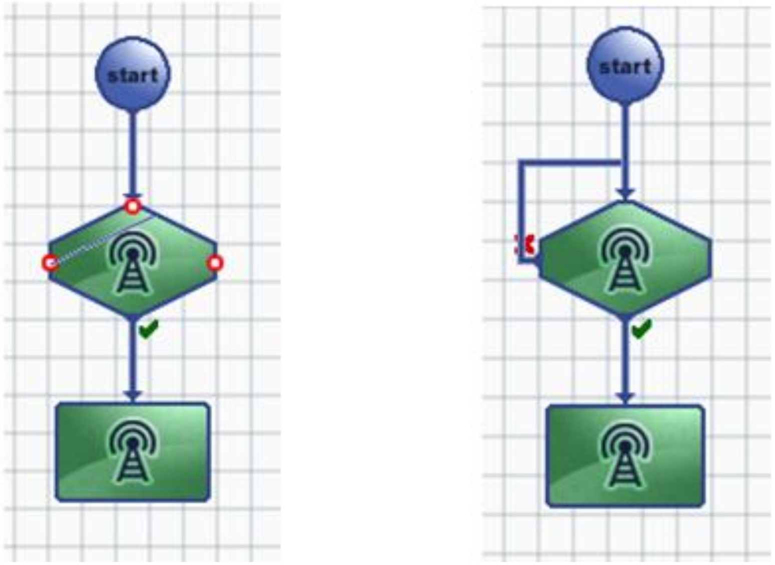


Image 32. Arrow drawing for a loop

The arrow track can be modified dragging the marks that appear when the cursor is over the arrow.

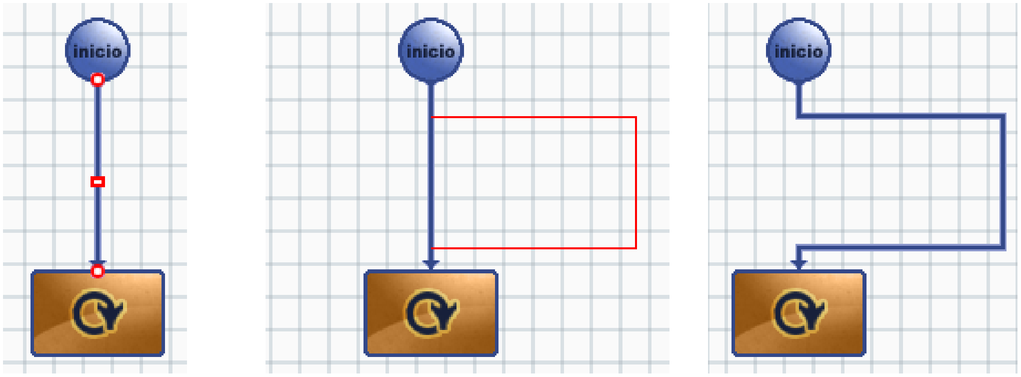


Image 33. Modifying arrow track

5.1.7. Language change and updates

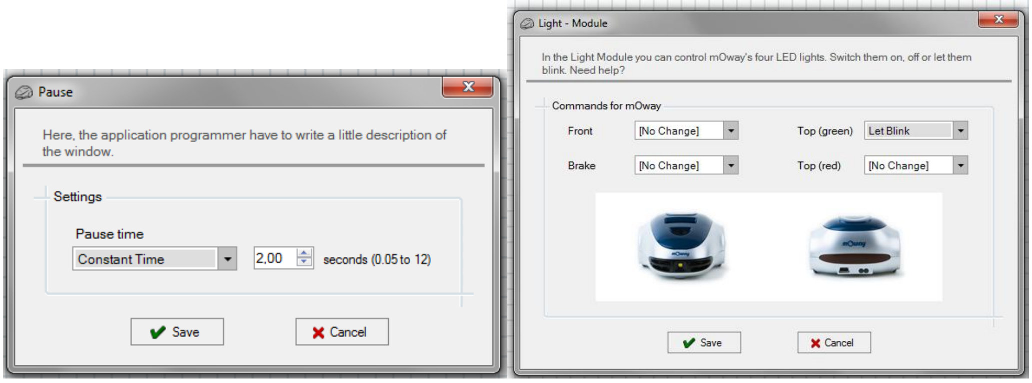
The language of MowayWorld can be changed from toolbar "Language" tab. It is also possible to check if MowayWorld needs to be updated in "Help – Check for updates".

5.2. *First program in MowayWorld*

This first basic program will make mOway avoid obstacles. Once MowayWorld is launched, save this project as “first_program”.

1. Once the project is saved, the program starts with a 2-second delay. Just drag and drop a “Pause” module and configure it with a constant value of 2 seconds.

2. The command to make green LED blink is added with the “Lights” module. Configuration of both modules are shown below.



Pause

Here, the application programmer have to write a little description of the window.

Settings

Pause time

Constant Time | 2,00 | seconds (0.05 to 12)

Light - Module

In the Light Module you can control mOway's four LED lights. Switch them on, off or let them blink. Need help?

Commands for mOway

Front: [No Change]
Top (green): Let Blink
Brake: [No Change]
Top (red): [No Change]

Image 34. Pause and LED configuration

3. The end of the program is added (“Finish” module) so that the application can be compiled.

4. The program is compiled and recorded into the robot clicking on “Program mOway” button in the toolbox, shown in the next image.

5. Test the program and check that after waiting 2 second the green LED blinks.

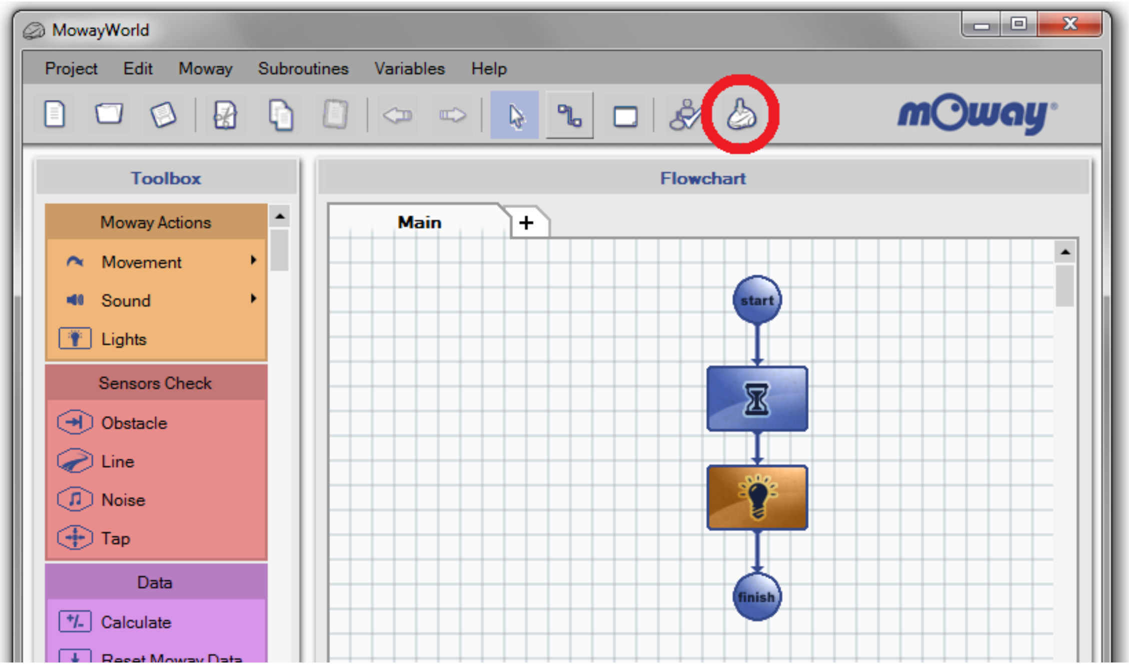
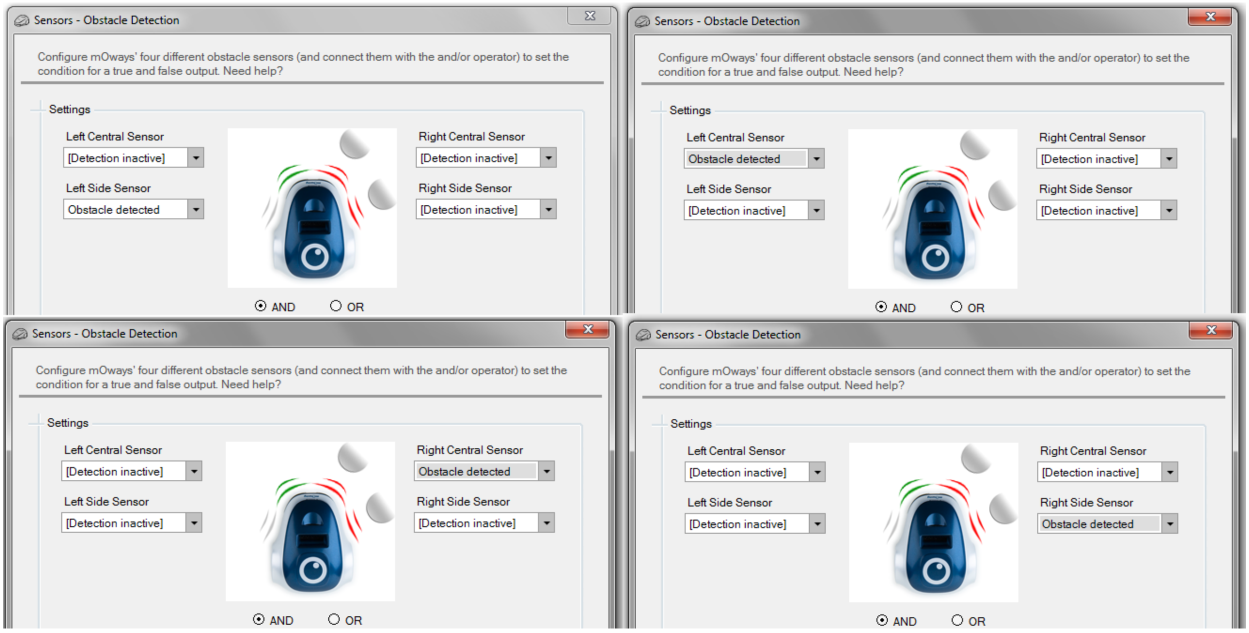


Image 35. Flowchart of the program and “Program mOway” button

6. In order to detect obstacles, four “Sensor Check Obstacle” has to be added and configured one for each obstacle sensor.


Sensors - Obstacle Detection

Configure mOways' four different obstacle sensors (and connect them with the and/or operator) to set the condition for a true and false output. Need help?

Settings

Left Central Sensor: [Detection inactive]
Left Side Sensor: Obstacle detected
Right Central Sensor: [Detection inactive]
Right Side Sensor: [Detection inactive]
● AND ○ OR

Sensors - Obstacle Detection

Configure mOways' four different obstacle sensors (and connect them with the and/or operator) to set the condition for a true and false output. Need help?

Settings

Left Central Sensor: Obstacle detected
Left Side Sensor: [Detection inactive]
Right Central Sensor: [Detection inactive]
Right Side Sensor: [Detection inactive]
● AND ○ OR

Sensors - Obstacle Detection

Configure mOways' four different obstacle sensors (and connect them with the and/or operator) to set the condition for a true and false output. Need help?

Settings

Left Central Sensor: [Detection inactive]
Left Side Sensor: [Detection inactive]
Right Central Sensor: Obstacle detected
Right Side Sensor: [Detection inactive]
● AND ○ OR

Sensors - Obstacle Detection

Configure mOways' four different obstacle sensors (and connect them with the and/or operator) to set the condition for a true and false output. Need help?

Settings

Left Central Sensor: [Detection inactive]
Left Side Sensor: [Detection inactive]
Right Central Sensor: [Detection inactive]
Right Side Sensor: Obstacle detected
● AND ○ OR

Image 36. Check Obstacles configuration

7. Condition modules have a true output and false output. When the obstacle is detected, the condition is true (green mark) and the corresponding LED turns on. If the obstacle is not detected, the condition is false (red mark) and the corresponding LED turns off. LED control is done with "Lights" modules.

8. Test the program and check that the front LEDs light up when an obstacle is detected.

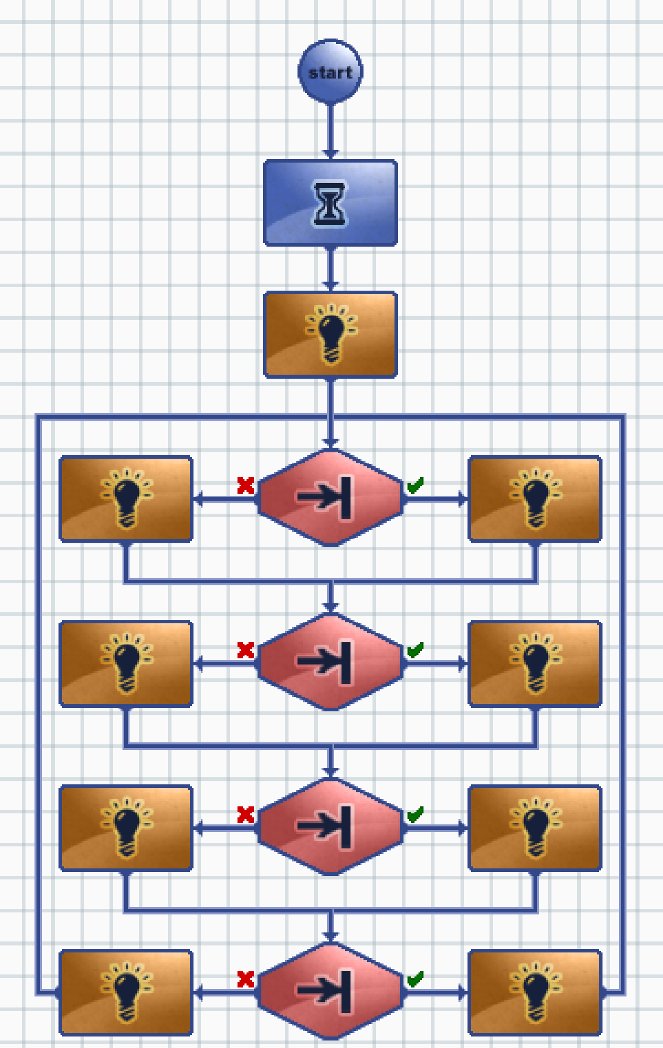
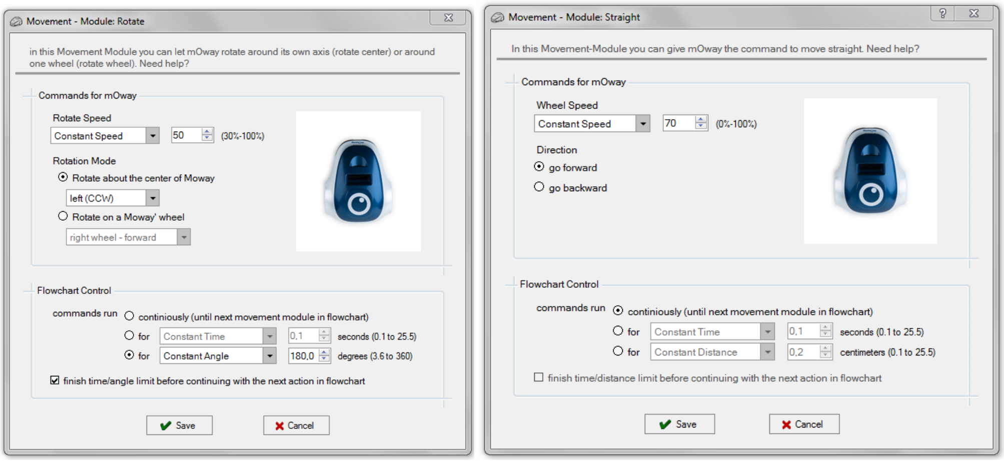


Image 37. First MowayWorld program: obstacle detection

9. We add an straight forward movement indefinitely until an obstacle is found.

10. When an obstacle is found, a command is sent to the robot to rotate 180º. The robot will continue to move in a straight line when the rotation is completed.



Movement - Module: Rotate

in this Movement Module you can let mOway rotate around its own axis (rotate center) or around one wheel (rotate wheel). Need help?

Commands for mOway

Rotate Speed: Constant Speed 50 (30%-100%)

Rotation Mode
- ● Rotate about the center of Moway: left (CCW)
- ○ Rotate on a Moway' wheel: right wheel - forward

Flowchart Control

commands run
- ○ continiously (until next movement module in flowchart)
- ○ for Constant Time 0,1 seconds (0.1 to 25.5)
- ● for Constant Angle 180,0 degrees (3.6 to 360)

☑ finish time/angle limit before continuing with the next action in flowchart

Movement - Module: Straight

In this Movement-Module you can give mOway the command to move straight. Need help?

Commands for mOway

Wheel Speed: Constant Speed 70 (0%-100%)

Direction
- ● go forward
- ○ go backward

Flowchart Control

commands run
- ● continiously (until next movement module in flowchart)
- ○ for Constant Time 0,1 seconds (0.1 to 25.5)
- ○ for Constant Distance 0,2 centimeters (0.1 to 25.5)

☐ finish time/distance limit before continuing with the next action in flowchart

Image 38. Movement and Rotation configuration

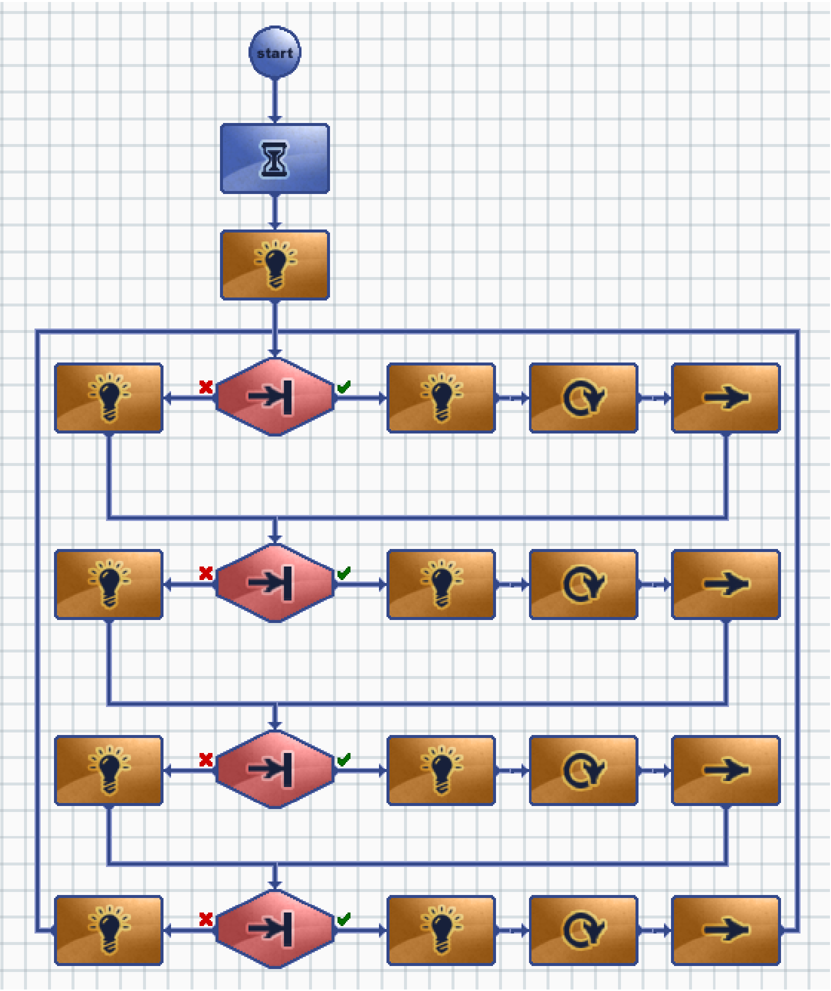


Image 39. End of first program in MowayWorld

5.3. Modules

The mOway programming consist of functions or actions that control the robot. Modules are these functions or actions that mOway can carry out: moving, reading sensors, transmission of radiofrequency messages, etc. These modules are grouped depending on their function.

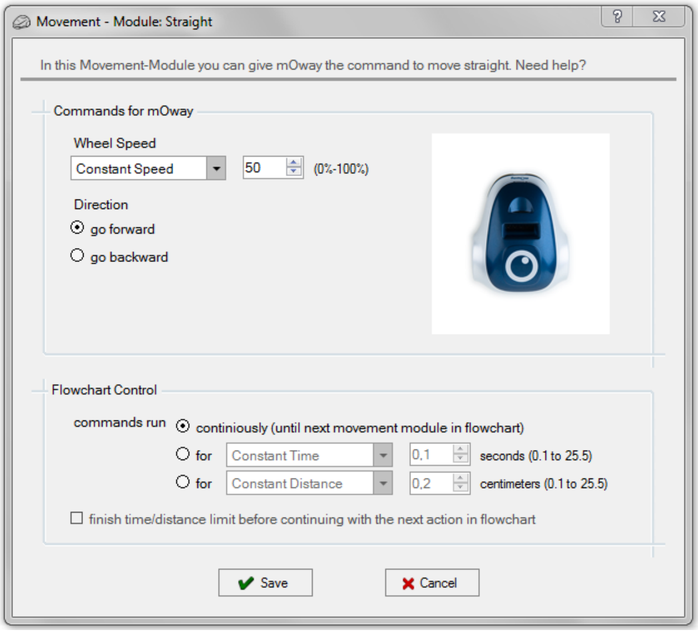
Each of the modules included in MowayWorld are described below.

5.3.1. Moway Actions

This group of modules makes it possible to control mOway's actuators: motors, speaker and LEDs. Functions of these modules can be performed for a user-defined period of time or distance if the **"finish time/distance limit before continuing with the next action in flowchart"** option is selected. If this option is not selected, the function will be executed indefinitely until another module changes the current function.

- **Movement – Straight**

mOway robot has two motors, one in each wheel. These give it a great flexibility in its movements. Movement straight command makes mOway to go forward or backward describing a straight line trajectory. Speed value can be modified.



Movement - Module: Straight

In this Movement-Module you can give mOway the command to move straight. Need help?

Commands for mOway

Wheel Speed

Constant Speed 50 (0%-100%)

Direction

● go forward

○ go backward

Flowchart Control

commands run ● continiously (until next movement module in flowchart)

○ for Constant Time 0,1 seconds (0.1 to 25.5)

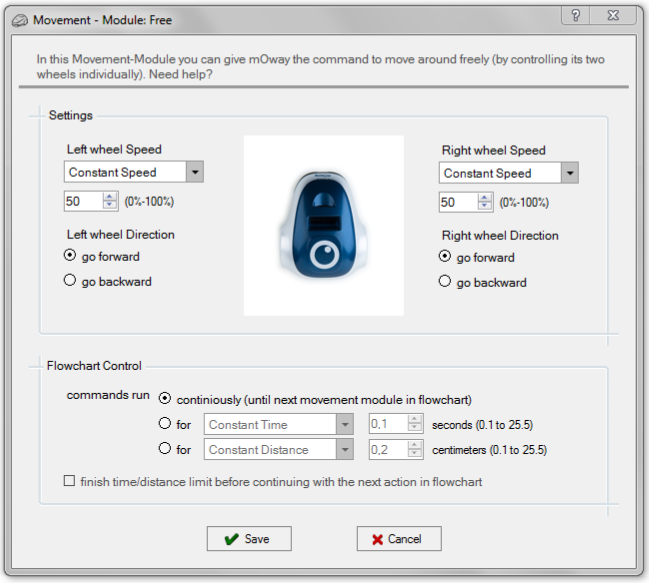
○ for Constant Distance 0,2 centimeters (0.1 to 25.5)

□ finish time/distance limit before continuing with the next action in flowchart

Image 40. Movement – Straight configuration window

- **Movement – Free**

Free movement is similar to straight movement, but in this case the speed of each motor can be adjusted separately.



Movement - Module: Free

In this Movement-Module you can give mOway the command to move around freely (by controlling its two wheels individually). Need help?

Settings

Left wheel Speed: Constant Speed; 50 (0%-100%)

Left wheel Direction: ● go forward ○ go backward

Right wheel Speed: Constant Speed; 50 (0%-100%)

Right wheel Direction: ● go forward ○ go backward

Flowchart Control

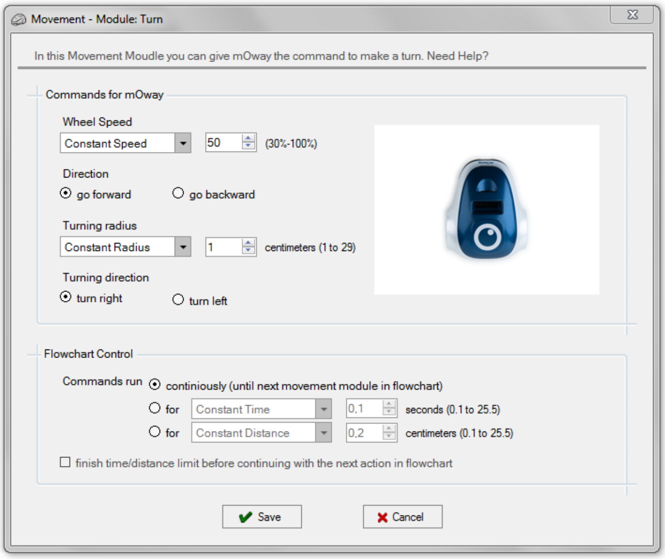
commands run ● continiously (until next movement module in flowchart)
○ for Constant Time 0,1 seconds (0.1 to 25.5)
○ for Constant Distance 0,2 centimeters (0.1 to 25.5)

□ finish time/distance limit before continuing with the next action in flowchart

Image 41. Movement – Free configuration window

- **Movement – Turn**

In this function, drive system will calculate the speed of the motors in order to be able to trace a curve, indicating the speed and turning curvature.



Movement - Module: Turn

In this Movement Moudle you can give mOway the command to make a turn. Need Help?

Commands for mOway

Wheel Speed: Constant Speed; 50 (30%-100%)

Direction: ● go forward ○ go backward

Turning radius: Constant Radius; 1 centimeters (1 to 29)

Turning direction: ● turn right ○ turn left

Flowchart Control

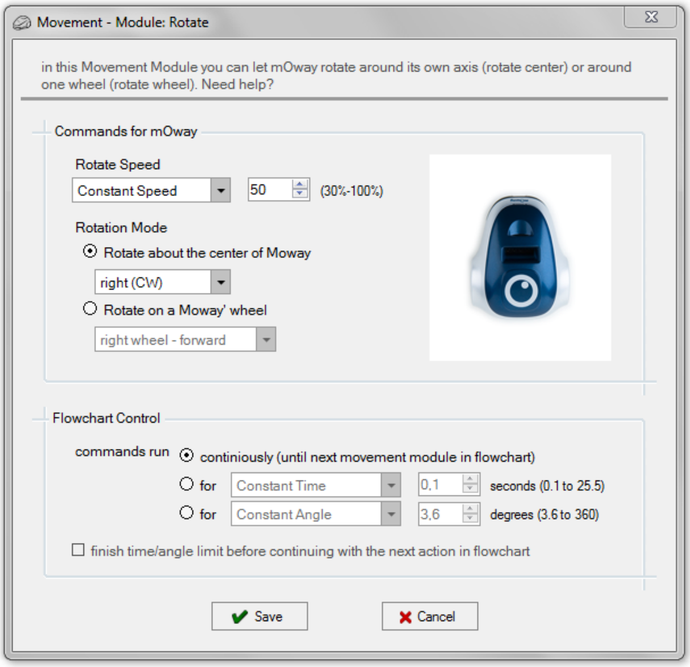
Commands run ● continiously (until next movement module in flowchart)
○ for Constant Time 0,1 seconds (0.1 to 25.5)
○ for Constant Distance 0,2 centimeters (0.1 to 25.5)

□ finish time/distance limit before continuing with the next action in flowchart

Image 42. Movement – Turn configuration window

- **Movement – Rotate**

With this command, mOway will rotate either on its centre or on one of its two wheels. Turning direction and rotation speed can be defined.



Movement - Module: Rotate

in this Movement Module you can let mOway rotate around its own axis (rotate center) or around one wheel (rotate wheel). Need help?

Commands for mOway

Rotate Speed

Constant Speed ▾ 50 (30%-100%)

Rotation Mode

● Rotate about the center of Moway

right (CW) ▾

○ Rotate on a Moway' wheel

right wheel - forward ▾

Flowchart Control

commands run ● continiously (until next movement module in flowchart)

○ for Constant Time ▾ 0,1 seconds (0.1 to 25.5)

○ for Constant Angle ▾ 3,6 degrees (3.6 to 360)

□ finish time/angle limit before continuing with the next action in flowchart

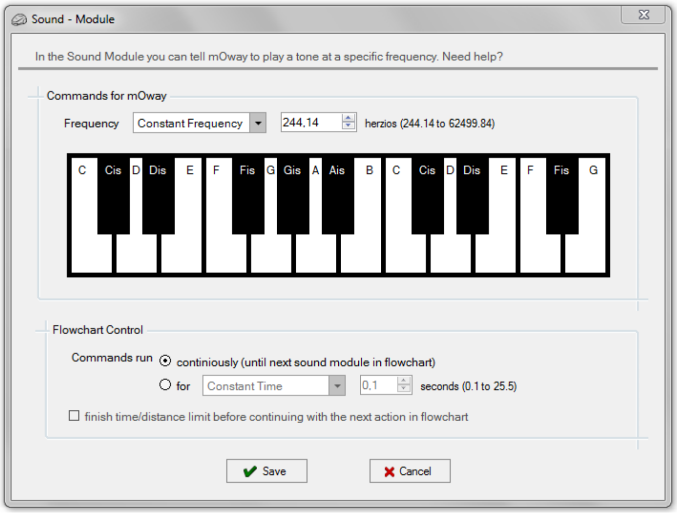
Image 43. Movement – Rotate configuration window

- **Movement – Stop**

This command stops mOway's motors.

- **Sound – Play**

This function makes mOway to emit tones from 244 Hz to 16 KHz. It is possible to define the time that the speaker will be activated.


Sound - Module

In the Sound Module you can tell mOway to play a tone at a specific frequency. Need help?

Commands for mOway

Frequency: Constant Frequency, 244.14 herzios (244.14 to 62499.84)

Flowchart Control

Commands run ● continiously (until next sound module in flowchart)

○ for Constant Time 0.1 seconds (0.1 to 25.5)

□ finish time/distance limit before continuing with the next action in flowchart

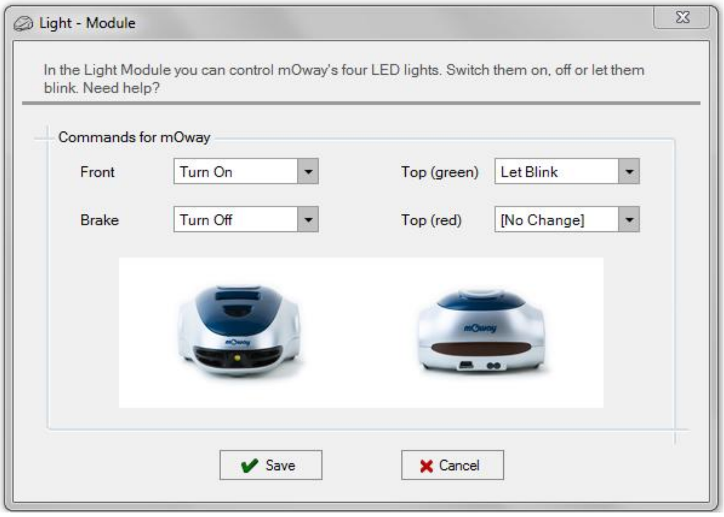
Image 44. Sound – Play configuration window

- **Sound – Stop**

Stops the speaker sound.

- **Lights**

This module makes it possible to operate on mOway's LED diodes. You can turn them on, turn them off or make them blink.


Light - Module

In the Light Module you can control mOway's four LED lights. Switch them on, off or let them blink. Need help?

Commands for mOway

Front: Turn On

Top (green): Let Blink

Brake: Turn Off

Top (red): [No Change]

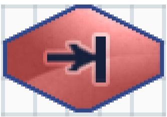
Image 45. Lights configuration window

5.3.2. Sensors Check

This group of modules makes it possible to get the mOway's sensors values. These are conditional modules, so that they have two different outputs: if the condition configured in the module is true, the output will be "true" (green mark on the flowchart). Otherwise, if the condition is false, the output will be "false" (red cross on the flowchart).

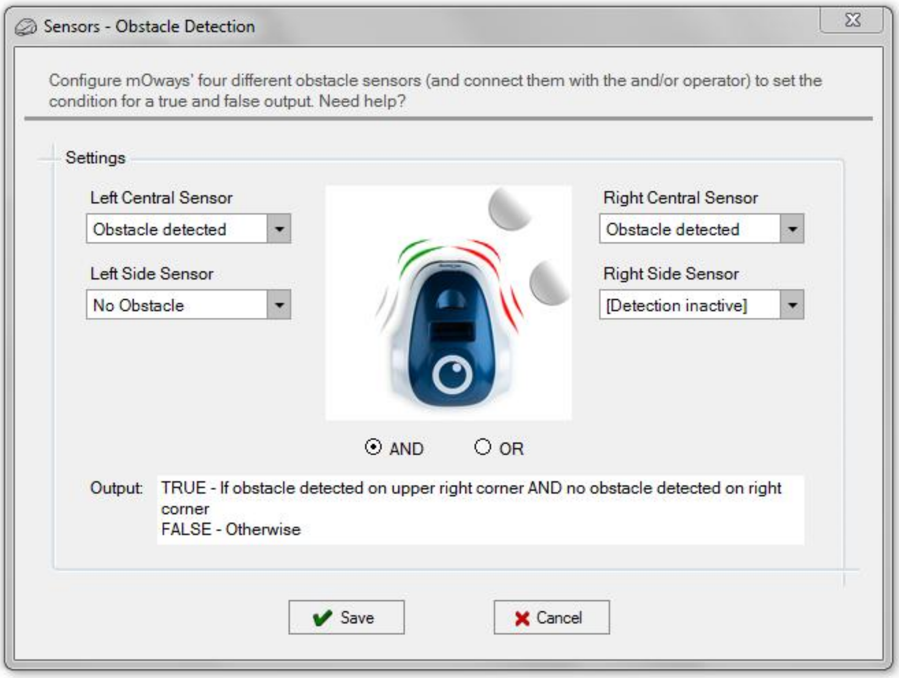
- **Obstacle**

This module checks the digital value of the four obstacle sensors. Each sensor can perform one of the following conditions:



- obstacle detected
- no obstacle detected
- detection inactive

In addition, it makes it possible to check the AND or OR boolean operation. If AND option is checked, all the conditions must be true to get a "true" output. On the other hand, if OR is checked, just one of all the condition has to be true to get a "true" output.



Sensors - Obstacle Detection

Configure mOways' four different obstacle sensors (and connect them with the and/or operator) to set the condition for a true and false output. Need help?

Settings

Left Central Sensor: Obstacle detected

Left Side Sensor: No Obstacle

Right Central Sensor: Obstacle detected

Right Side Sensor: [Detection inactive]

● AND ○ OR

Output: TRUE - If obstacle detected on upper right corner AND no obstacle detected on right corner
FALSE - Otherwise

Image 46. Sensors Check - Obstacles configuration window

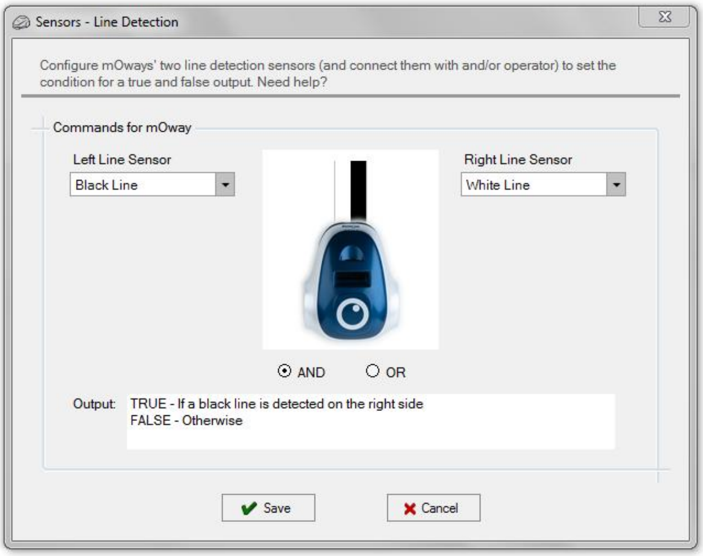
- **Line**

This module checks the digital value of line sensors. This module is very useful for making mOway follow a line (black or white) on the floor, detect boundaries, etc.

Each sensor can perform one of the following conditions:

- black line detection
- white line detection
- detection inactive

In addition, it makes it possible to check the AND or OR boolean operation. If AND option is checked, all the conditions must be true to get a "true" output. On the other hand, if OR is checked, just one of all the condition has to be true to get a "true" output.



Configure mOways' two line detection sensors (and connect them with and/or operator) to set the condition for a true and false output. Need help?

Commands for mOway

Left Line Sensor: Black Line

Right Line Sensor: White Line

● AND ○ OR

Output: TRUE - If a black line is detected on the right side
FALSE - Otherwise

Image 47. Sensors Check – Line configuration window

- **Noise**

This module has a "true" output if mOway detects a loud sound. It has not any configuration window.

- **Tap**

This module has a "true" output if mOway detects a tap or a high acceleration. It has not any configuration window.

5.3.3. Data

This group of modules make it possible to read, write and modify data that mOway robot can provide from sensors and internal memory.

Data is stored in user-defined variables. A variable is created by clicking on “Variable” on the toolbar and then clicking on “New variable”. A name and an initial value have to be defined for this new variable (see “Variables” chapter).

Some of the modules allow to create these variables directly from their configuration window. This permits to configure the module with values that can be modified during the execution of the program.

NOTE: Each variable is stored in one byte of mOway’s internal memory, which means that the value of the variable can be from 0 to 255.

- **Calculate**

This module adds (+) or subtract (-) two values. First parameter is always a variable and it stores the operation result. Second parameter can be a constant or a variable.

- **Reset mOway Data**

This module initializes time and distance counters stored in mOway’s internal memory.

- **Assign Variable**

This group of modules makes it possible to assign a value to a user-defined variable. This value can be constant or the analog value of mOway’s sensors.

  - **Value:** assigns a constant value to a variable.
  - **Time:** assigns elapsed time to a variable. Value goes from 0 to 255 tenths of a second.
  - **Speed:** assigns the speed value of one of the wheels to a variable. Value goes from 0 (stopped) to 100 (maximum speed).
  - **Distance:** assigns distance value covered by mOway to a

variable. Value goes from 0 to 255 centimetres.

- **Angle:** assigns the value of mOway turning angle to a variable. Value goes from 0 (0 degrees) to 100 (corresponding to 360 degrees). Only changes with “Rotate” module.
- **Brightness:** assigns the light sensor value to a variable. Value goes from 0 (dark) to 100 (light).
- **Line:** assigns one of the line sensors value to a variable. Value goes from 0 (white colour detection) to 100 (black colour detection).
- **Obstacle:** assigns one of the obstacle sensors value to a variable. Value goes from 0 (no detection) to 100 (closest detection).
- **Accelerometer:** assigns one of accelerometer axis value to a variable. Value goes from 0 (negative acceleration limit) to 255 (positive acceleration limit). When there is not acceleration, the value is 127 (range mid-point).
- **Noise:** assigns the microphone value to a variable. Value goes from 0 (silence) to 100 (loud noise).
- **Temperature:** assigns the robot temperature value to a variable. Value goes from 0ºC to 255ºC.
- **Battery:** assigns battery level to a variable. Value goes from 0% to 100%.

- **Compare**

This group of modules makes it possible to compare a variable or a sensor value with another value (constant value or user-variable). The following are the comparison operators: equal (==), different (<>), greater (>), greater or equal (>=), less (<), less or equal (<=).

- **Data:** compares a variable with a constant value or a user-variable.
- **Time:** compares elapsed time. Values goes from 0.1 to 25.5 seconds.
- **Speed:** compares one of the wheels speed value. Values

goes from (stopped) to 100 (maximum speed).

- **Distance:** compares distance value covered by mOway to a variable. Value goes from 0 to 255 centimetres.
- **Angle:** compares the value of mOway turning angle. Value goes from 0 (0 degrees) to 100 (corresponding to 360 degrees). Only changes with “Rotate” module.
- **Brightness:** compares the light sensor value. Value goes from 0 (dark) to 100 (light).
- **Line:** compares one of the line sensors value to a variable. Value goes from 0 (white colour detection) to 100 (black colour detection).
- **Obstacle:** compares one of the obstacle sensors value to a variable. Value goes from 0 (no detection) to 100 (closest detection).
- **Accelerometer:** compares one of accelerometer axis value. Value goes from -2g (negative acceleration limit) to 2g (positive acceleration limit). When there is not acceleration, the value is 0 (NOTE: *g* value is 9.81m/s^2).
- **Noise:** compares microphone value. Value goes from 0 (silence) to 100 (loud noise).
- **Temperature:** compares the robot temperature value. Value goes from 0ºC to 255ºC.
- **Battery:** compares battery level. Value goes from 0% to 100%.

5.3.4. Flowchart Control

- **Pause**

This module makes it possible to insert a pause in the program with a duration set in multiples of 0.05 seconds. The pause parameter may be a constant or a variable.

- **Call Function**

This module makes it possible to call a function or subroutine defined by user. Subroutines will be explained on chapter "Functions /Subroutines".

- **Finish**

This module sets the program end. If the program consist of an infinite loop, this module is not necessary.

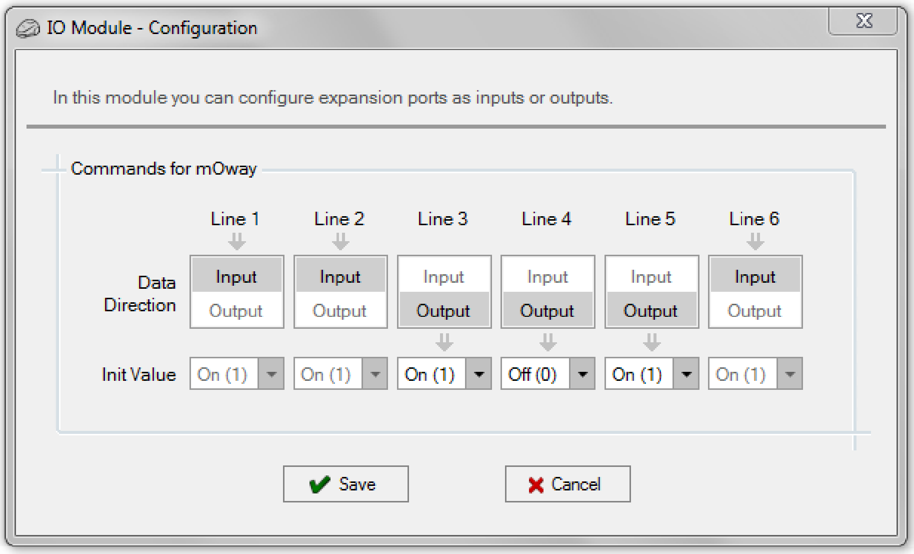
5.3.5. Expansion

This group of functions makes it possible to control the expansion connector of mOway robot, either to use radiofrequency (RF) communication, mOway camera module or user expansion modules.

<u>WARNING!</u>
Only advanced users can use the pinout configuration. Any incorrect connection of electronic elements to the expansion connector may damage the robot irreversibly or the user circuitry.

- **Module IO – Configuration**

This module configures the expansion connector pins as inputs or outputs and assigns a initial value for outputs.



IO Module - Configuration

In this module you can configure expansion ports as inputs or outputs.

Commands for mOway

	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6
Data Direction	Input	Input	Output	Output	Output	Input
Init Value	On (1)	On (1)	On (1)	Off (0)	On (1)	On (1)

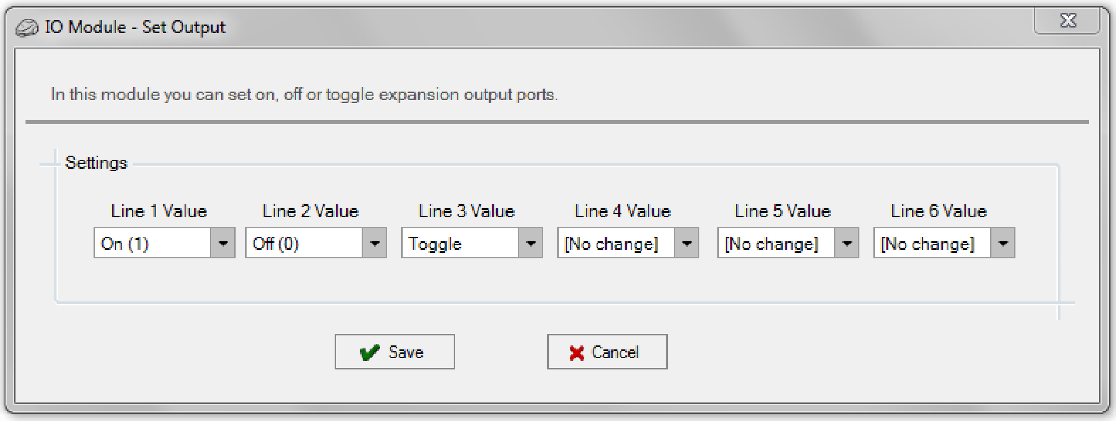
Image 48. Module IO – Configuration configuration window

- **Module IO – Set Output**

This module sets (On), resets (Off) or toggles expansion connector pins that are configured as outputs.

WARNING!

"Set Output" values only can be assigned for pins configured as outputs, leaving other lines with "No change" value. If a value is assigned to an input, this may damage the robot irreversibly.



IO Module - Set Output

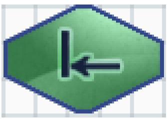
In this module you can set on, off or toggle expansion output ports.

Settings

Line 1 Value	Line 2 Value	Line 3 Value	Line 4 Value	Line 5 Value	Line 6 Value
On (1)	Off (0)	Toggle	[No change]	[No change]	[No change]

Image 49. Module IO – Set Output configuration window

- **Module IO – Check Input**

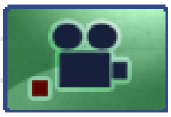


This module checks the digital value of one of the 6 pins of the expansion connector. It compares if the value of the selected pin is equal (==) or different (<>) to “on” (1) or “off” (0).

- **Camera – Play**

This module activates the mOway camera. One of the four channels has to be selected. This channel must be the same of the **Moway Camera Board** video receptor.

- **Camera – Stop**

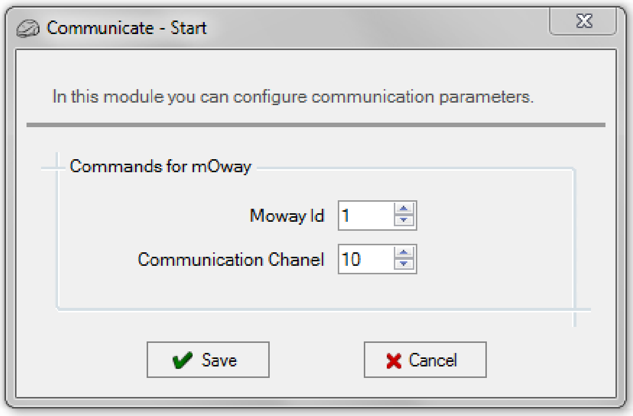


This module turns the camera off.

- **Communicate – Start**

This module configures the robot in order to establish a radiofrequency (RF) communication. An identification for the robot and a communication channel must be selected.

For establishing communication between two mOways, they must be configured with the same channel and an unique address for each one. If the configurations is correct, the output is “true”. Otherwise, the output is “false”.



Communicate - Start

In this module you can configure communication parameters.

Commands for mOway

Moway Id: 1

Communication Chanel: 10

Image 50. Communicate – Start configuration window

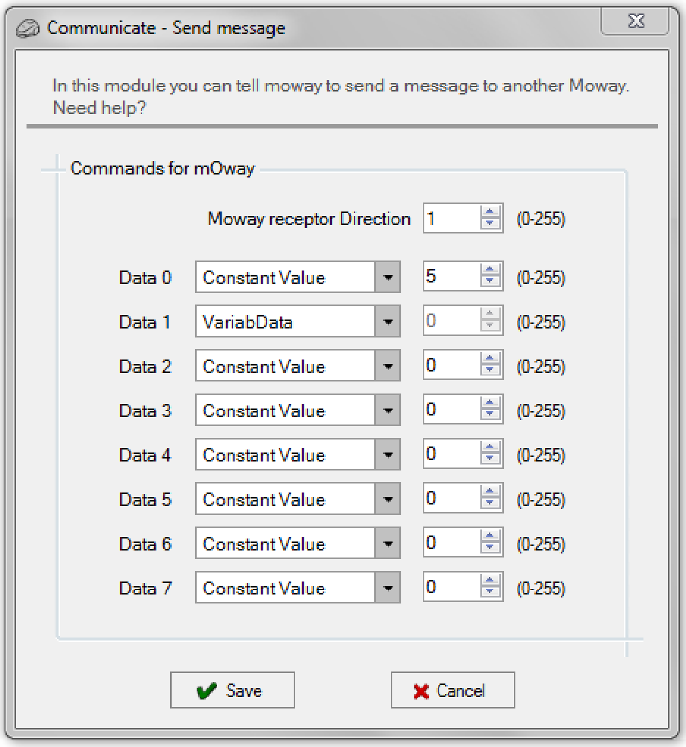
- **Communicate – Stop**

This module stops the RF communication. It has not any configuration window.

- **Communicate – Send**

This module makes it possible to transmit a frame to a specific address. The address of the receptor and the data, which can consists of constants or variables, must be indicated in the frame. If the sent data has been received by the receptor, the output is "true". Otherwise the output is "false".

Before using this conditional, the RF communication must be configured using the "Communicate - Start" module. All the robots taking part in the RF communication must have the same channel and different addresses.



Communicate - Send message

In this module you can tell moway to send a message to another Moway.
Need help?

Commands for mOway

Moway receptor Direction 1 (0-255)

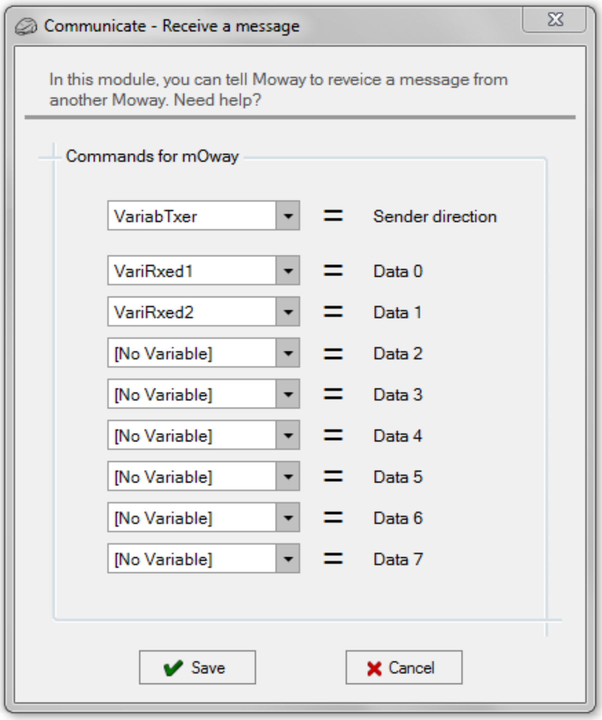
Data 0	Constant Value	5	(0-255)
Data 1	VariabData	0	(0-255)
Data 2	Constant Value	0	(0-255)
Data 3	Constant Value	0	(0-255)
Data 4	Constant Value	0	(0-255)
Data 5	Constant Value	0	(0-255)
Data 6	Constant Value	0	(0-255)
Data 7	Constant Value	0	(0-255)

Image 51. Communicate – Send configuration window

- **Communicate – Receive**

This module makes it possible to receive a frame from a specific address. It must be indicated at least two variables: one for collecting the transmitter address and the other for the data. If the data sent by the transmitter has been received correctly, the output is “true”. Otherwise the output is “false”.

Before using this conditional, the RF communication must be configured using the "Communicate - Start" module. All the robots taking part in the RF communication must have the same channel and different addresses.



Communicate - Receive a message

In this module, you can tell Moway to reveice a message from another Moway. Need help?

Commands for mOway

VariabTxer = Sender direction
VariRxed1 = Data 0
VariRxed2 = Data 1
[No Variable] = Data 2
[No Variable] = Data 3
[No Variable] = Data 4
[No Variable] = Data 5
[No Variable] = Data 6
[No Variable] = Data 7

Image 52. Communicate – Receive configuration window

5.4. Variables

For more flexible programs some modules make it possible to use variables. Variables makes possible to save values into robot internal memory. A variable contains a value that can change during the execution of the program.

This is useful to read the values of the mOway's internal sensors (for example, speed value, pause time, read analog obstacle sensor values, etc.). Then these values can be compared, transmitted through RF to check sensor status, etc.

Variables can be created from some modules. They can also be defined from the toolbar, by clicking on "Variables" and then clicking on "New variable". A name and an initial value have to be declared. The name must not have more than 14 letters.

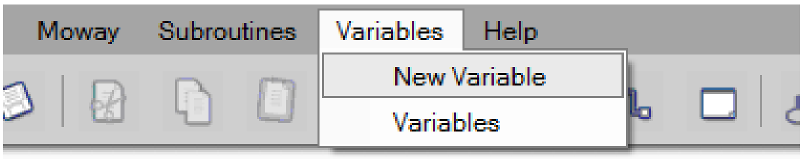
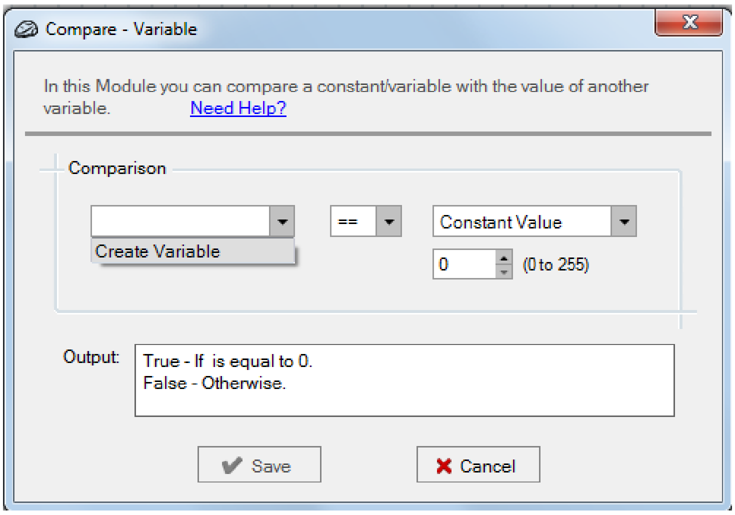


Image 53. Creation of a new variable from toolbar


Compare - Variable

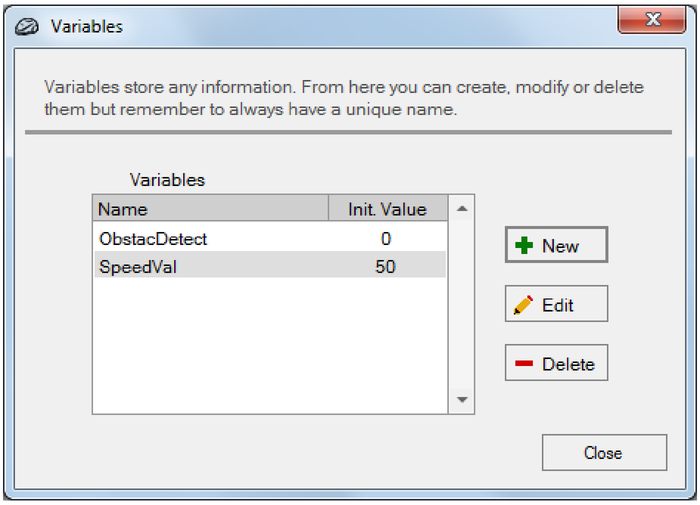
In this Module you can compare a constant/variable with the value of another variable. Need Help?

Comparison

Output: True - If is equal to 0.
False - Otherwise.

Image 54. Creation of a new variable from a module

Variables can be edited from toolbar, “Variables – Variables”.



Variables store any information. From here you can create, modify or delete them but remember to always have a unique name.

Variables

Name	Init. Value
ObstacDetect	0
SpeedVal	50

Image 55. Variables window

5.5. *Functions / Subroutines*

A function or subroutine is a piece of program that can be used in another program. For example, a user can develop a diagram that turns on the front LED, waits one second, then turns off the LED and finally waits one more second.

If this program is defined as a Function, it can be called from the main diagram. In order to create a new function, click on “+” icon and give it a name (for example, “MyFunction”). A new tab called “MyFunction” will appear on the top of the flowchart editor.

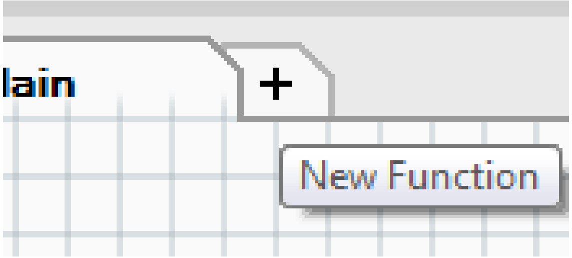


Image 56. Creation of a new function

The functionality described above can be developed in the “MyFunction” flowchart editor.

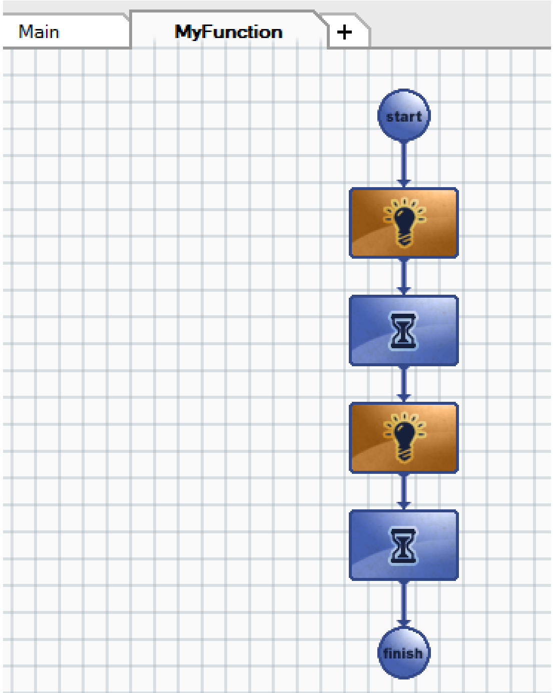


Image 57. MyFunction diagram

In order to go back to the main diagram, click on “Main” tab. There the “MyFunction” function can be called (“Flowchart Control – Call Function”). In the “Call Function” configuration window, “MyFunction” is selected.

In this example, it is called three times, so that the front LED will blink three times, once per second.

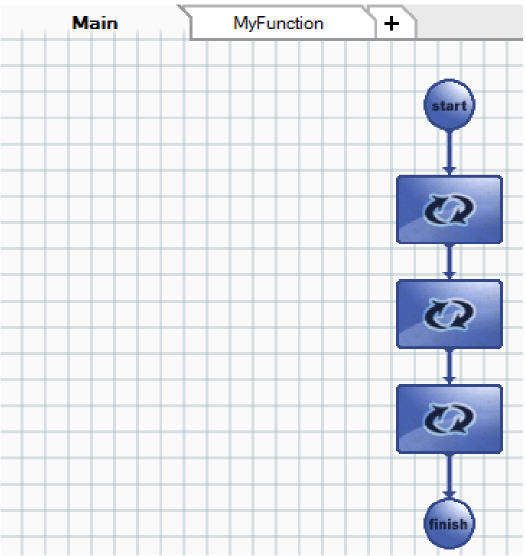


Image 58. Calling “MyFunction” three times from main diagram

6. Applications

6.1. Communications Window

The Communications Window makes it possible to send and receive RF messages using RFUSB device connected to PC. In order to start RF communication, press the “Communications” icon.

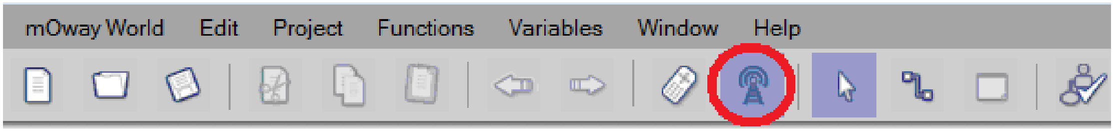
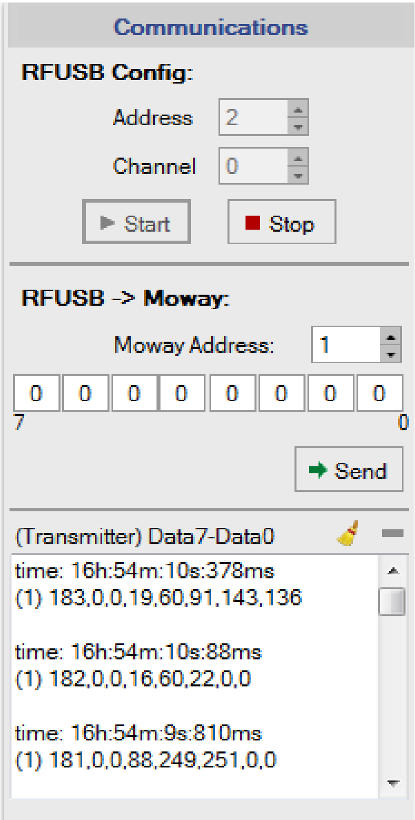


Image 59. “Communications” icon

The Communications Window will appear on the right side of the workspace. In order to start RF communication, choose an address and a RF channel for the RFUSB (“2” and “0” in this example) and press **Start** button.



RFUSB Config:

Address: 2

Channel: 0

RFUSB -> Moway:

Moway Address: 1

0	0	0	0	0	0	0	0
7							0

(Transmitter) Data7-Data0

```
time: 16h:54m:10s:378ms
(1) 183,0,0,19,60,91,143,136

time: 16h:54m:10s:88ms
(1) 182,0,0,16,60,22,0,0

time: 16h:54m:9s:810ms
(1) 181,0,0,88,249,251,0,0
```

Image 60. “Communications” window

In order to send a message from PC to mOway, select mOway robot address (“1” in this example), write the data to be sent (from Data 7 to Data 0) and press **Send** button. If a “0” is selected for “Moway Address” the message will be received by all the robots near the PC.

When RFUSB receives a message from mOway, the communication window displays the time when the message was received and the data. The format of the data is:

- Address of transmitter mOway is displayed in brackets
- Data arranged from Data 7 to Data 0

6.2. Moway Cam

MowayCam application displays the camera images and lets the user to save a static image in a storage device connected to the computer. This application can be launched from MowayWorld toolbar (“View -> mOway Cam) or clicking this icon:

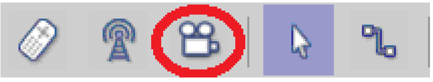
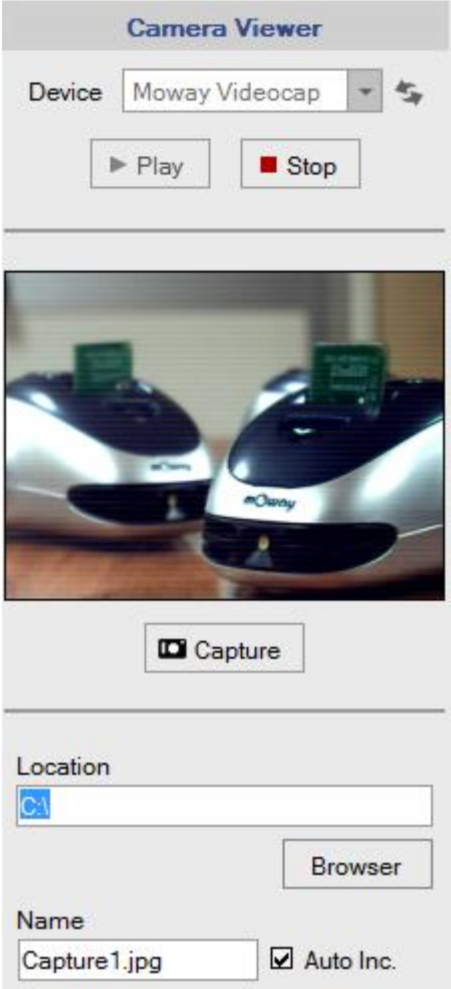


Image 61. MowayCam access from MowayWorld

Once the icon is pressed, MowayCam lateral panel appears.



Camera Viewer

Device: Moway Videocap

Location: C:\

Name: Capture1.jpg

☑ Auto Inc.

Image 62. MowayCam panel

This window displays camera images and allows the user to control the video receptor and to save a static image on PC.

- Click on **Refresh** button to see all the video devices connected to PC. Depending on the driver version, Moway video receptor may be named as “Moway Videocap”, “STK1160 Grabber” or “USB2.0 ATV”.
- Click on **Play** button to start displaying camera images.
- Click on **Stop** button to deactivate video receptor **Moway Camera Board**.
- Click on **Save** button to save the current image in the path and name displayed on that fields.

IMPORTANT: Video receptor **Moway Camera Board** MUST NOT be disconnected from USB port while MowayCam is showing camera images. If it is disconnected while showing camera images, some computers could restart. Click on Stop button or close MowayCam panel before disconnecting it.

6.3. MowayRC

MowayRC is an application to control mOway as if it was a radio control device and to monitor all the robot's sensors. This tool, which uses RF BZI-RF2GH4 modules and RFUSB (mOway Base is compatible), is very useful for all those users wishing to explore the field where the microbot will perform.

Its functioning concept is as follows: the application transmits commands by means of the USB to the RFUSB, which transmits them to mOway, where a recorded program interprets those commands (Moway_RC_Client included in Moway Pack).

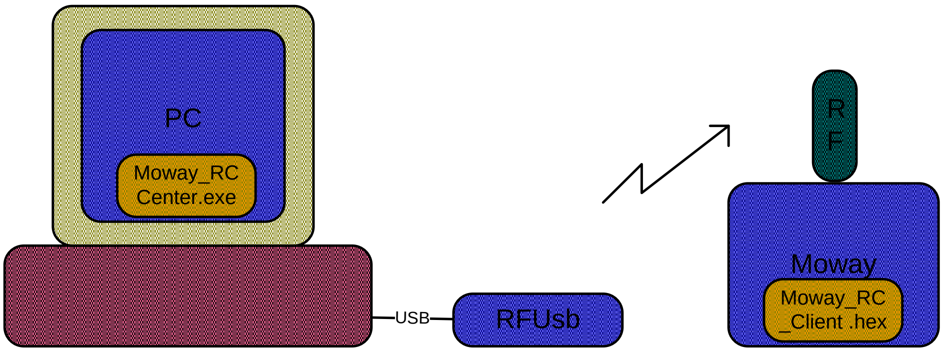
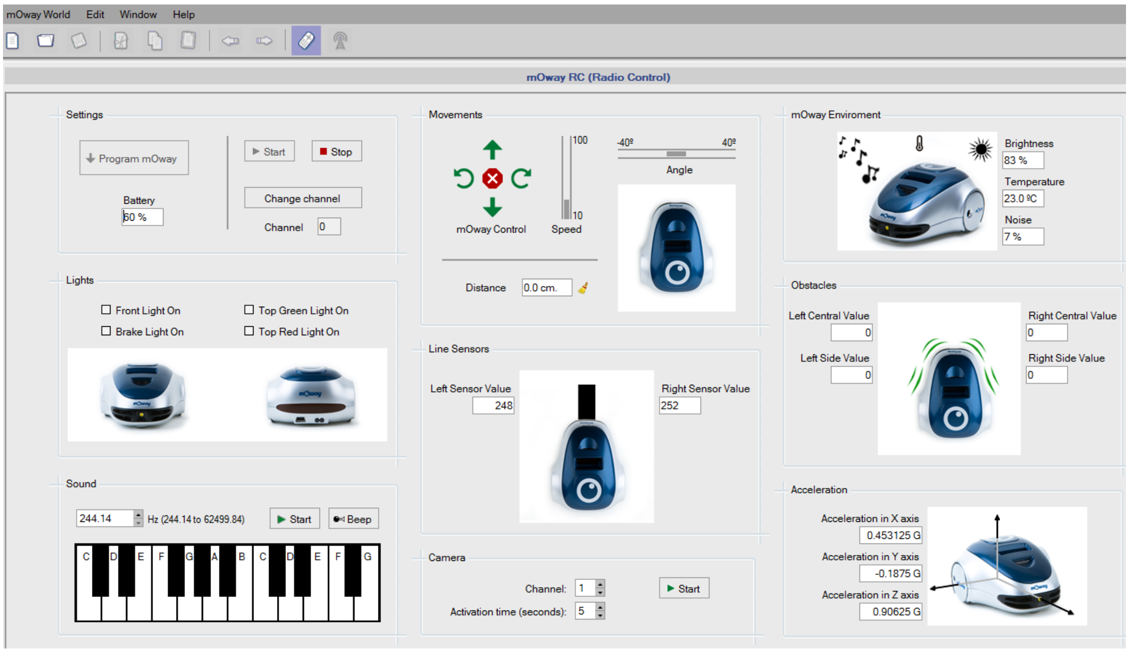


Image 63. Moway RC Diagram



Battery: 60 %

Channel: 0

Distance: 0.0 cm.

Brightness: 83 %

Temperature: 23.0 ºC

Noise: 7 %

Left Central Value: 0

Right Central Value: 0

Left Side Value: 0

Right Side Value: 0

Left Sensor Value: 248

Right Sensor Value: 252

244.14 Hz (244.14 to 62499.84)

Channel: 1

Activation time (seconds): 5

Acceleration in X axis: 0.453125 G

Acceleration in Y axis: -0.1875 G

Acceleration in Z axis: 0.90625 G

Image 64. Moway RC Center

You can access the application in “View – Moway RC”. The program “mOway_RC_Client” is downloaded in the robot and the application starts. The fields of the RC Center will be explained on the next lines.

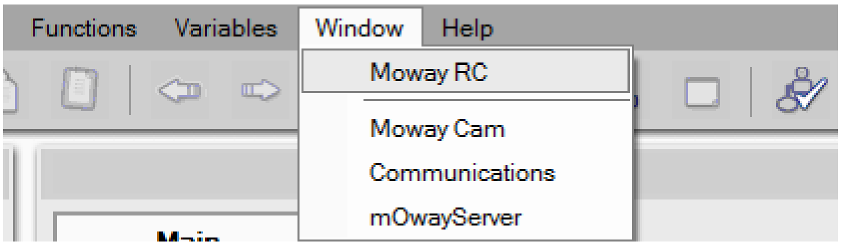


Image 65. Access to Moway RC

6.3.1. *RF configuration*

In this part the RFUSB module is configured with a “2” default address and “0” for the communications channel (default mOway RC program channel).

Once connected, the communications channel can be changed if WI-FI, Bluetooth, Microwaves, other mOways, etc. interferences are detected in this first channel. Click on the change channel button to select up to 16 channels. To change the channel the robot has to be switched on and be in communication with the RFUSB. Every time the RFUSB is disconnected the default channel shall be “0”.

The recommended procedure is as follows:

1) Turn the robot on
2) Connect the *RFUSB*
3) Test the channel sending mOway commands
4) If the robot does not react well change the channel and try again

In case the robot is not programmed with “mOway_RC_Client”, it can be programmed pressing **Program mOway** button.

6.3.2. *Movements*

Once the RFUSB is connected mOway can be sent commands. The robot’s movements can be controlled by means of the panel buttons or keyboard (*W*, *S*, *A*, *D* keys). There also are two bars to determine the speed and turning curvature. The covered distance can be reset with **Reset distance** button.

6.3.3. *LED*

In this section mOway’s four LEDs are switched on and off.

6.3.4. *Speaker*

In this section is checked the switched on and off of the robot´s speaker in a particular frequency. He robot can also reproduce music notes.

6.3.5. *Sensor status*

This section describes the values returned by the sensors at all times (updated every second).

- Line sensors: the darker is the surface, the higher is the value.
- mOway environment: brightness, temperature and noise detected by the robot.
- Obstacles: the closer is the obstacle, the higher is the value.
- Acceleration: 3 axis acceleration values.

6.3.6. *Camera*

It activates mOway camera on the selected channel for a maximum of 15 seconds. While the camera is on, the communication between mOway and PC stops, in order to avoid interferences. Once the activation time finishes, the communication automatically starts again.

In order to watch camera images it is necessary to launch “Camera” panel.

6.4. *MowayServer*

This application allows mOway to communicate with other devices, such as phones, tables and PCs through Wifi technology. These devices can connect to the net established by mOway by means of **Moway Wifi Module**[15]. Once they are connected, the robot can be controlled from device web browser.

The application is launched from toolbar.

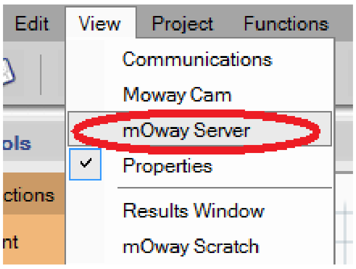
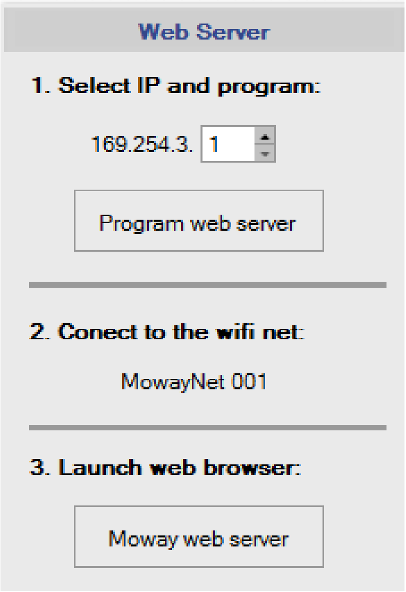


Image 66. Access to Moway Web Server

[15] Module not available in all kits

The lateral panel appears.



Web Server

1. Select IP and program:

169.254.3. 1

2. Conect to the wifi net:

MowayNet 001

3. Launch web browser:

Image 67. Moway Web Server panel

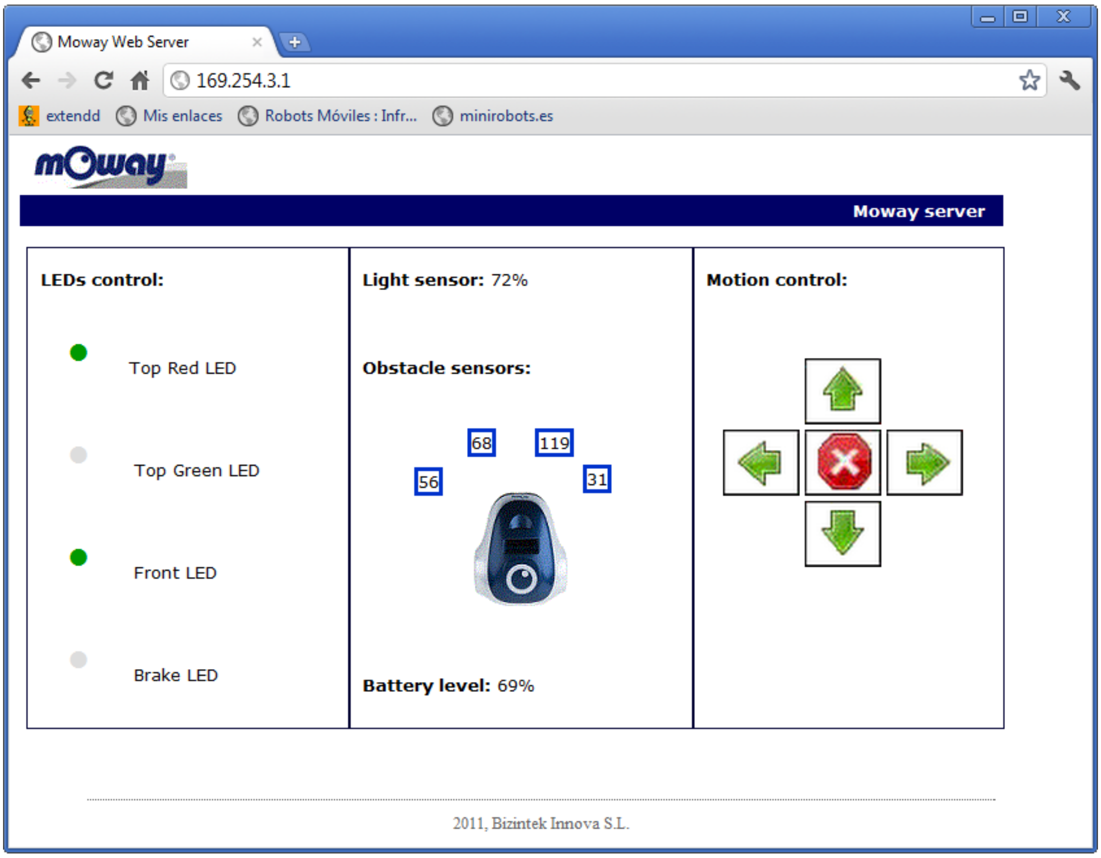
In order to communicate with mOway Web Server, it is necessary to follow these steps:

1) Select an IP (only the last field can be modified) and press **Program web server** button.

2) The robot will establish an adhoc wifi net with the name MowayNet*** (*** is the selected number for IP in the previous step).

3) If the web server will be accessed from PC, press **Moway web server** button to launch web browser with the selected IP. If another device is used, launch web browser and introduce the selected IP.

Once this is done, the mOway Web Server will be displayed. It makes it possible to control mOway movements, LEDs and read some sensor values.



LEDs control:

● Top Red LED

○ Top Green LED

● Front LED

○ Brake LED

Light sensor: 72%

Obstacle sensors:

Battery level: 69%

Motion control:

2011, Bizintek Innova S.L.

Image 68. A Moway Web Server

NOTE: Connecting to MowayNet or web server displaying can take a while, depending on the wifi connection and on the web browser.

7. Simulation

7.1. Introduction

MowayWorld simulator allows testing the working of a program before programming it in mOway robot. In this way you can detect any errors you could have in the program and fix them quickly.

The simulator is opened from the toolbar. NOTE: When you activate the simulator, the side panels remain inactive.

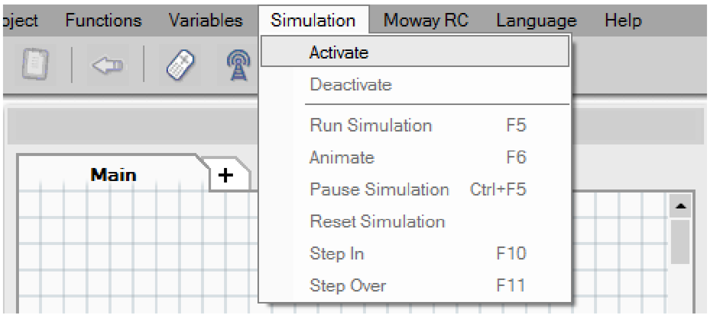
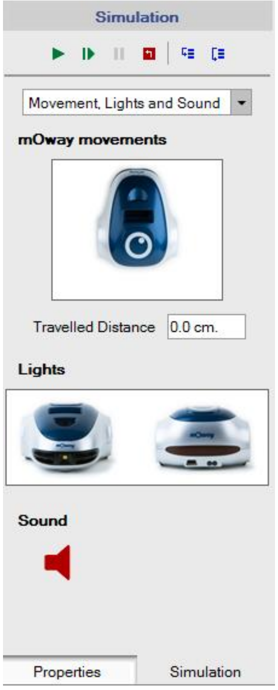


Image 69. Simulator access

By activating the simulator, it will appear in the side panel:



Movement, Lights and Sound

Travelled Distance 0.0 cm.

Image 70. Simulator side panel

The modes available in the simulator are the following:

Run: In *Run* mode, the simulator executes the program in “real time”, i.e., the speed at which the program would execute in the real robot.

Animate: In *Animate* mode the simulator executes the program step by step, pausing for a second in each block. The block that is running the simulator at each time is indicated by a yellow arrow.

Pause: Stops the simulation at the current block.

Reset: Pressing the reset button, the simulation returns to the initial state, i.e., the beginning of the program. Also resets the value of the variables.

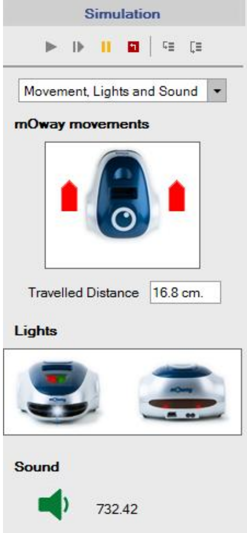
Step In: In this mode, the simulator moves through the program step by step, stopping at each block. It also moves through the functions created by the user.

Step Over: In this mode, the simulator moves through the program step by step stopping at each block. Unlike the previous mode, the simulator does not advance into the user functions.

7.2. Functioning

In the simulator the user can vary the mOway sensors value and see how this affects the operation of the robot, depending on the program it’s being simulated. The sections of the robot that can be viewed and controlled in the simulation are the following:

- **Movement, lights and sound**: shows the state of the motors, the state of the LEDs and the mOway speaker.
    - The rotation of the wheels is indicated by red arrows. It also reflects the distance travelled by the robot.
    - The sound emitted by the robot is indicated by the speaker icon. When it is green, it simulates that mOway beeps .When it is red, it simulates that the sound has stopped.



Simulation

Movement, Lights and Sound

mOway movements

Travelled Distance: 16.8 cm.

Lights

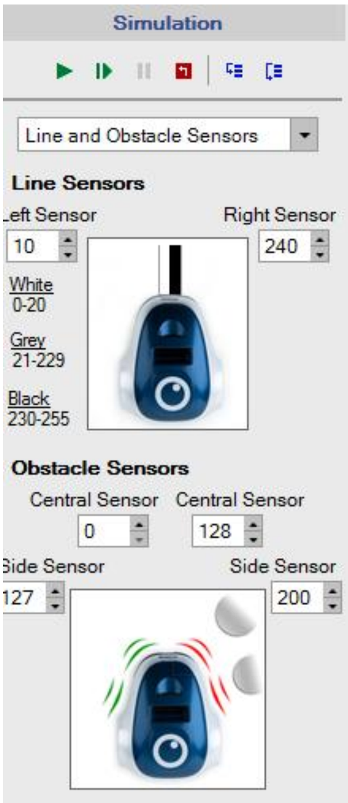
Sound

732.42

- **Line and obstacles sensors**: simulates the change of the value of the line and obstacles sensors. The value of the simulator sensors is defined by the analog value of the real sensors of the robot. Therefore, the value of the simulated sensor are the following:

Line sensors	
Simulated line	**Sensor value**
White	0 to 20
Grey	21 to 229
Black	230 to 255

Obstacle sensors	
Simulated obstacle	**Sensor value**
Obstacle not detected	0 to 127
Obstacle detected	128 to 255



Simulation

Line and Obstacle Sensors

Line Sensors

Left Sensor: 10

Right Sensor: 240

White 0-20

Grey 21-229

Black 230-255

Obstacle Sensors

Central Sensor: 0

Central Sensor: 128

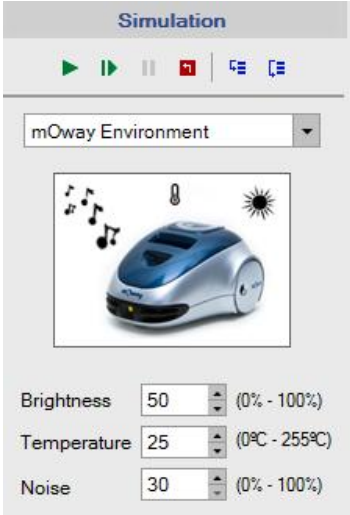
Side Sensor: 127

Side Sensor: 200

- **Environment**: it simulates the variation of the light and temperature sensors and the microphone.

Light sensor	
Simulated environment	**Sensor value**
Darkness	0
High-brightness	100

Noise sensor (microphone)	
Simulated environment	**Sensor value**
Silence	0
Loud noise	100



mOway Environment

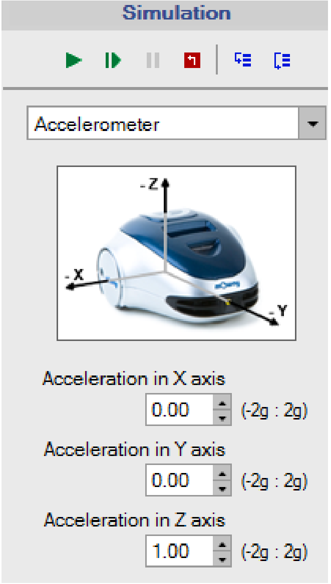
Brightness 50 (0% - 100%)

Temperature 25 (0ºC - 255ºC)

Noise 30 (0% - 100%)

- **Accelerometer**: it simulates the variation of the accelerometer value in its 3 axis. The value of the accelerometer in each axis varies between -2g and +2g, being "g" the acceleration of gravity (9,81m/s^2). For example, if we simulate that mOway is on an even surface, the values are the following:

Accelerometer	
Accelerometer axis	**Accelerometer value (g)**
X axis	0
Y axis	0
Z axis	1.00



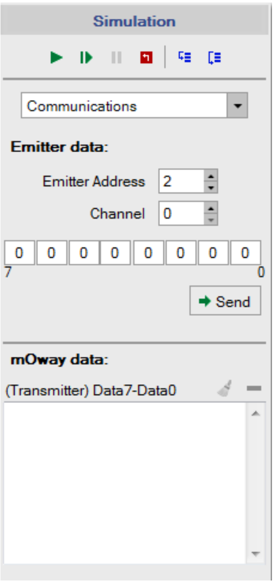
Accelerometer

Acceleration in X axis 0.00 (-2g : 2g)

Acceleration in Y axis 0.00 (-2g : 2g)

Acceleration in Z axis 1.00 (-2g : 2g)

- **Communications**: it simulates the transmitting of RF data to the mOway robot, and also the receiving data sent by the robot. “Emitter data” field simulates the data sending to the robot. “mOway data” field shows the data sent by the simulated robot.



Simulation

Communications

Emitter data:

Emitter Address 2

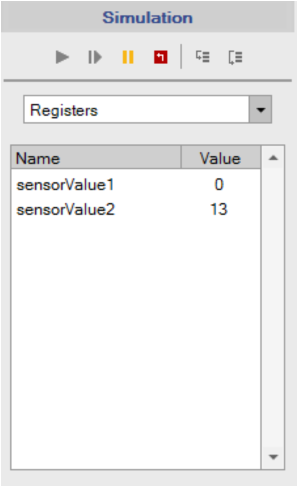
Channel 0

0	0	0	0	0	0	0	0

mOway data:

(Transmitter) Data7-Data0

- **Registers**: it shows a list of the variables created in the program with its values at all times.



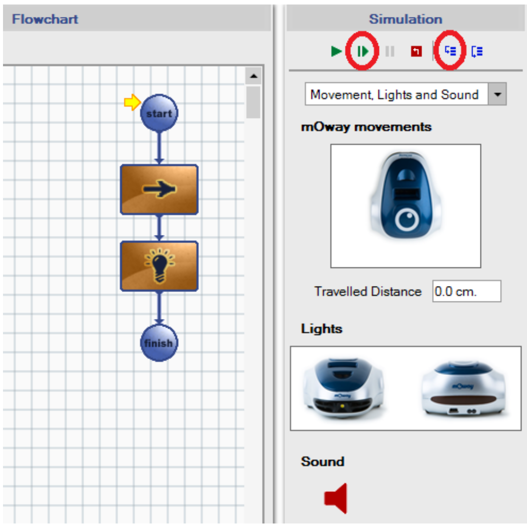
Simulation

Registers

Name	Value
sensorValue1	0
sensorValue2	13

7.3. *Simulation example*

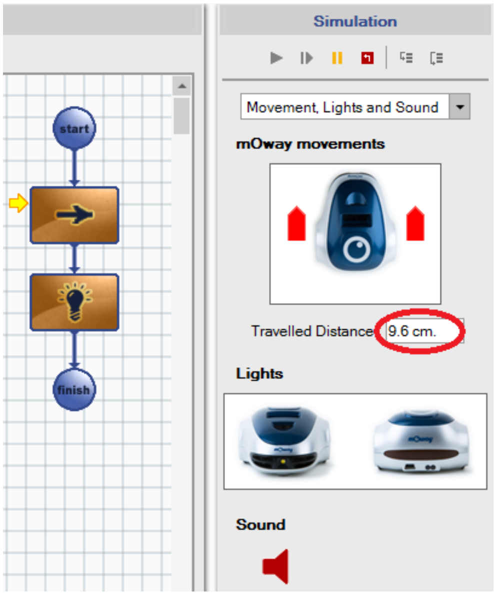
The following example shows the simulation of a program in which the robot moves straight for 12cm and then lights the front LED. Pressing the button "Animate" or "Step In", a yellow arrow indicates where the simulator is in the diagram.



Movement, Lights and Sound

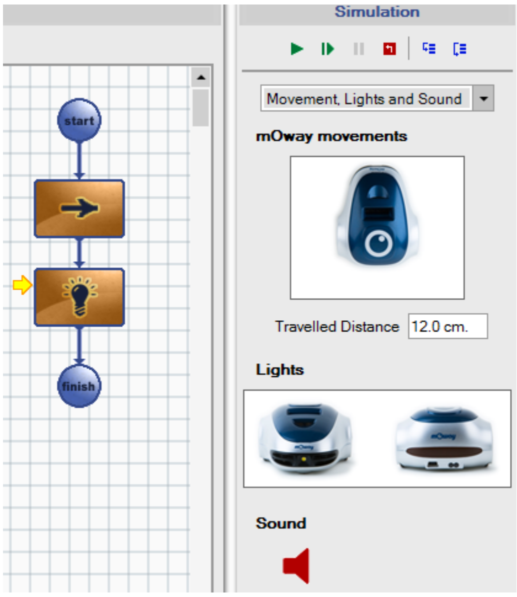
Travelled Distance 0.0 cm.

While the simulator executes the block "Straight", it indicates the direction of the rotation of the wheels, and the travelled distance is increased.



Movement, Lights and Sound

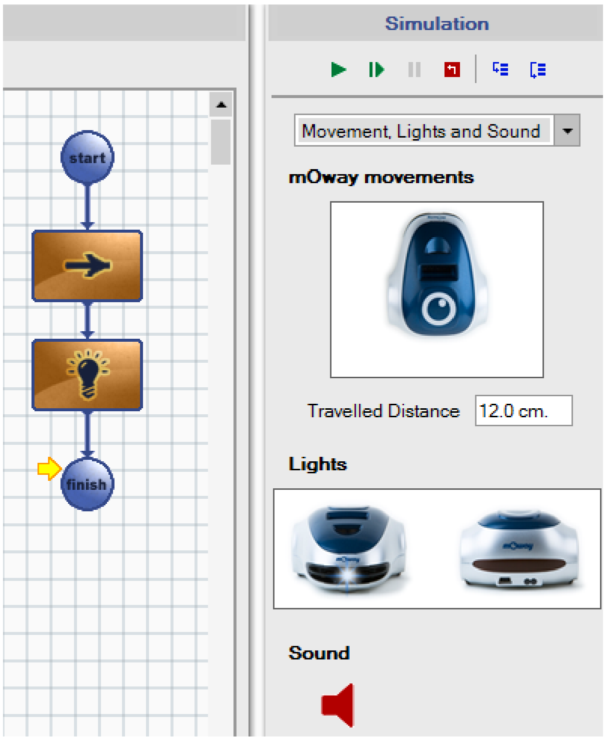
Travelled Distance 9.6 cm.



Movement, Lights and Sound

Travelled Distance 12.0 cm.

Once the block “Straight” has finished (the robot would travel 12cm),the simulator executes the block “Lights”, and the front LED lights. This done, the simulator ends.



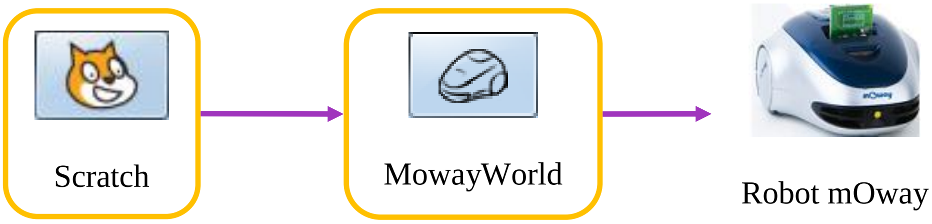
Movement, Lights and Sound

Travelled Distance 12.0 cm.

8. mOway Scratch

8.1. Introduction

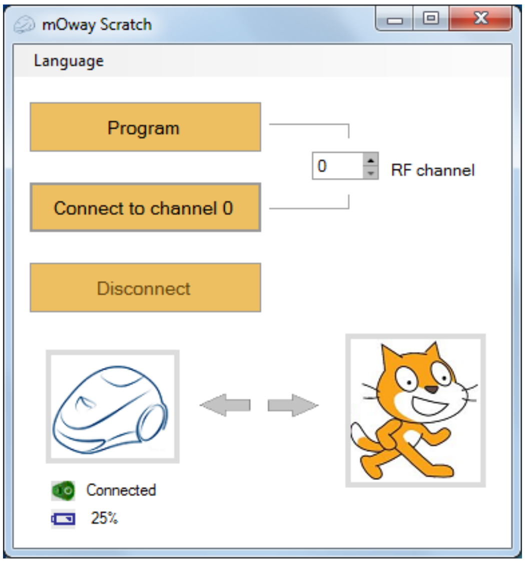
Now mOway is also compatible with Scratch. The communication between Scratch and mOway is bidirectional: Scratch receives the mOway detectors value, and at the same time mOway receives Scratch commands. This functionality uses Scratch sockets connection to obtain data from remote sensors and uses “send to all” commands for sending orders to mOway. In this case “mOway Scratch” application works as data gateway between Scratch and mOway.



The connection between mOway and Scratch is done by the mOway RF modules and the RFUSB module connected to the computer. <u>Therefore it is necessary having the RF module placed in the mOway expansion slot and the RFUSB module connected to a computer USB port.</u>

8.2. Functioning

The mOway-Scratch communication is activated by opening “mOway Scratch” application.



The “Program” button programs the needed firmware in mOway for a right connection with Scratch. In this case, mOway robot should be connected to a USB port. The firmware will be programmed to work in the selected channel in the “RF channel” box. Once the firmware is programmed in a certain channel, this channel should be selected when we get connected to Scratch.

To initiate communication, the “Connect to channel...” button is pressed. mOway robot should have the Scratch firmware previously programmed, the RFUSB module should be connected to PC and Scratch must be opened with a mOway sample program.

Once the connection from Scratch is started it is possible to obtain the value of any of the mOway sensors, and also it is possible to send moving orders to mOway, or switch on/off the LEDs and buzzer. To configure these actions we use variables in Scratch.

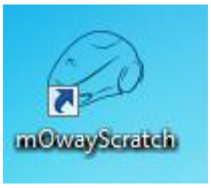
8.3. Step by step

1. Connect RF module into the expansion connector of mOway robot.

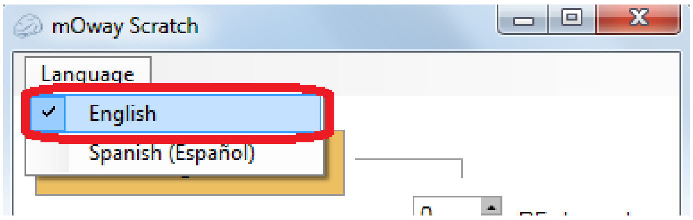
2. Connect the RFUSB to PC.

3. Connect mOway robot to PC.

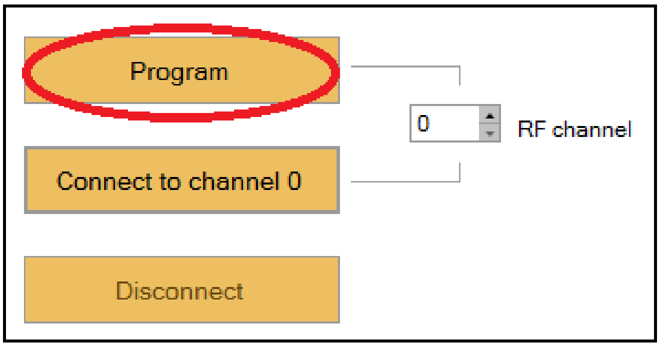
4. Open “mOway Scratch” application:



5. Select the language in which the Scratch programs will be developed (in this case, “English”).

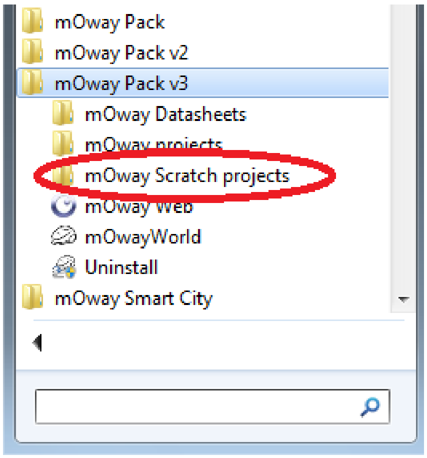


6. Press the “Program” button on the lateral panel (this is not necessary if the robot has been previously programmed in this way).

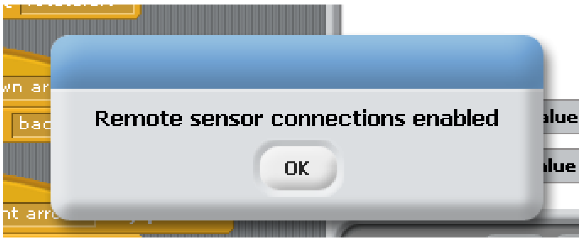


7. Once the robot has been programmed, disconnect it from the PC and switch the robot on.

8. Open one of the Scratch projects for mOway. You can open them form “*Start -> All programs -> mOway Pack v3 -> mOway Scratch projects*”. For example, open the “moway_RC.sb” project.

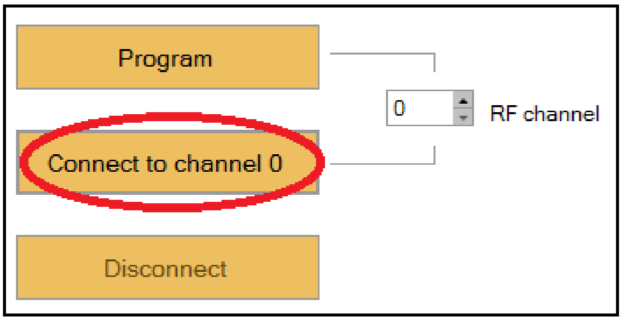


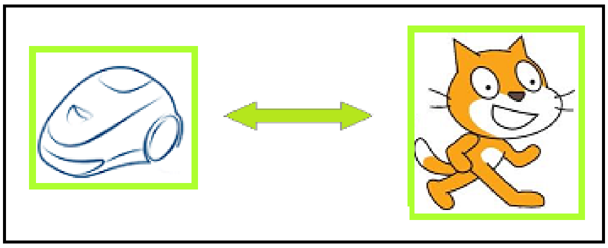
9. Press “OK”.



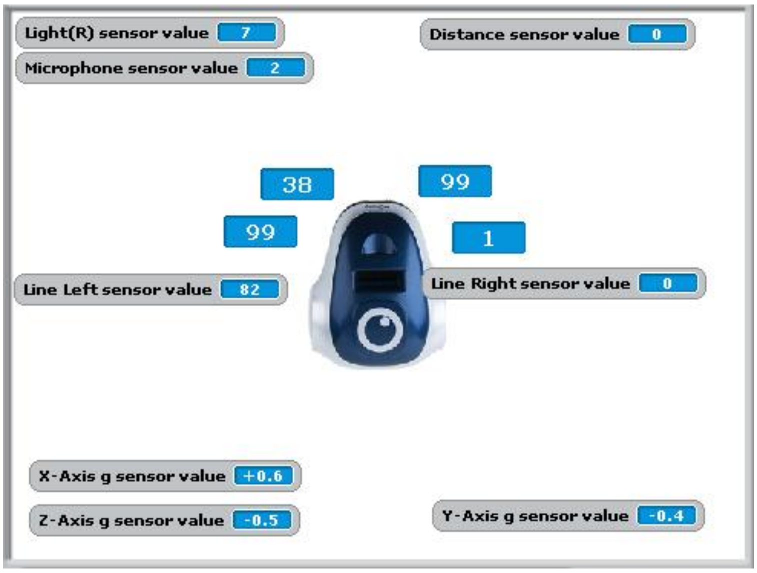
Remote sensor connections enabled

10. Once the Scratch project is open, press the “Start” button on the lateral panel of MowayWorld. The panel status is the following:



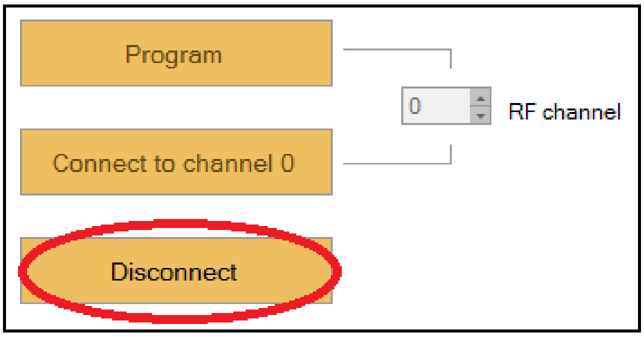


11. On the Scratch window you will see the sensor values. When the arrow keys of the keyboard are pressed, the robot will move.

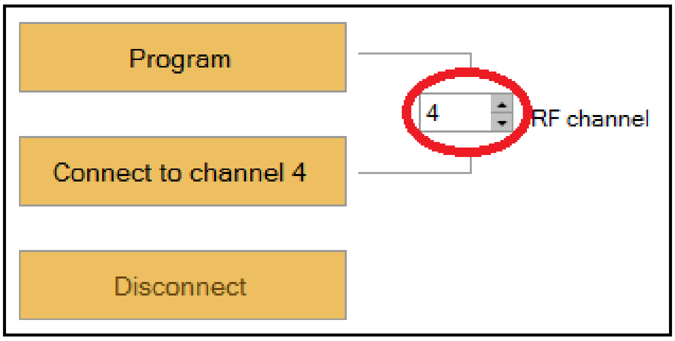

Light(R) sensor value 7
Microphone sensor value 2
Distance sensor value 0
38
99
99
1
Line Left sensor value 82
Line Right sensor value 0
X-Axis g sensor value +0.6
Z-Axis g sensor value -0.5
Y-Axis g sensor value -0.4

In case several robots are used in different PCs, it is necessary to assign a different channel for each one, in order to avoid interferences between them. These are the steps to follow:

1. If the communication with Scratch has been started, press the "Disconnect" button.



2. Assign a different channel to each robot (in this case, the channel 4 is selected):



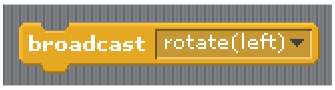
3. Reprogram the robot by pressing the “Program” button.

4. Once the robot has been programmed, disconnect it from the PC and switch the robot on.

5. Press the “Connect to channel...” button.

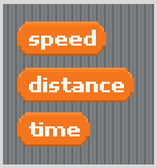
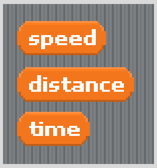
8.4. Commands and Sensors

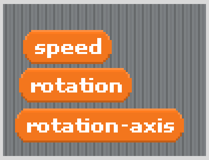
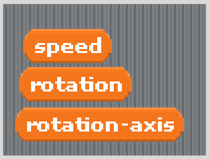
The actions available to run from the Scratch environment are displayed below, as well as the variables for each action. The command sending to mOway is done by using the control “broadcast” block.



broadcast rotate(left)

For example the message “rotate(left)” would make a left rotation.

Action	Description	Used variables
go	mOway goes straight forward indefinitely, with a speed of 50%.	-
back	mOway goes straight backward indefinitely, with a speed of 50%.	-
left	mOway turns 90 degrees left.	-
right	mOway turns 90 degrees right.	-
turnaround	mOway turns around.	-
stop	mOway stops.	-
reset(distance)	To reset the total distance counter motors.	-
go(straight)	mOway goes straight at a certain speed for the “distance” distance or “time” time. If “time” and “distance” are same than zero, mOway will move indefinitely.	 speed distance time
back(straight)	mOway goes back straight. It’s behaviour is identical to “go(straight)” but in the opposite way.	 speed distance time

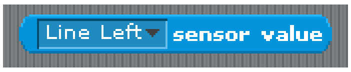
go(left)	mOway moves in a curve to the left of a certain radius at a certain speed for the distance “distance” or the time “time”. If “time” and “distance” are same than zero, mOway will move indefinitely.	
go(right)	mOway moves in a curve to the right. Identical but in the opposite direction of “go(left)”.	
back(left)	mOway goes back in a curve to the left. Identical to “go(left)” but going backwards.	
back(right)	mOway goes back in a curve to the right. Identical but in the opposite direction “back(left)”	
rotate(left)	mOway rotates at a certain speed in a certain angle to the left, on its axis or on a wheel. If the angle (rotation) is same than zero it rotates indefinitely.	
rotate(right)	mOway rotates to the right. Identical but in the opposite direction “rotate(left)”.	
frontled(on)	Activates the front LED.	-
frontled(off)	Deactivates the front LED.	-
frontled(blink)	Blinks the front LED.	-
brakeled(on)	Activates the brake LED.	-
brakeled(off)	Deactivates the brake LED.	-
brakeled(blink)	Blinks the brake LED.	-
greenled(on)	Activates the green LED.	-
greenled(off)	Deactivates the green LED.	-
greenled(blink)	Blinks the green LED.	-
redled(on)	Activates the red top LED.	-
redled(off)	Deactivates the red top LED.	-
redled(blink)	Blinks the red top LED.	-
leds(on)	Activates all the LEDs.	-

leds(off)	Deactivates all the LEDs.	-
leds(blink)	Blinks all the LEDs.	-
buzzer(on)	Activates the buzzer with a sound in a certain frequency.	frequency
buzzer(off)	To deactivate the buzzer.	-
enclosed	Subprogram for enclosing mOway into a black circle. It goes forward when it is on a white surface and it turns around when it reaches a black line.	-
push	Subprogram for pushing objects. mOway searches the objects and pushes them.	-
defender	Subprogram for pushing objects out of a black circle. It puts together the “enclosed” and “push” subprograms.	-
linefollow(left)	Subprogram for following a line on the left side.	-
linefollow(right)	Subprogram for following a line on the right side.	-

The following table shows the variables and the values can be taken for the correct execution of the commands:

Variable	Description	Range of values
distance	Distance to cover in mm	0 – 255 mm (0 – 25,5 cm)
frequency	Signal frequency speaker	0 – 16000 Hz
radius	Radius of curvature	0 – 100 (radius + speed < 100)
rotation	Rotation angle in degrees	0 – 360 º
rotation-axis	Rotation axis	Wheel – over a wheel Any value – over the center
speed	Movement speed	30 – 100 %
time	Movement time in tenths of a second	0 – 255 (0 - 25,5 seconds)

The following table shows the mOway variables we can read in Scratch using the "sensor value" block.



For example the previous block indicates the remaining battery level of mOway.

Variable	Description	Range of values
Obstacle Side Left	Obstacle Side Left Sensor	0 – 255
Obstacle Center Left	Obstacle Center Left	0 – 255
Obstacle Center Right	Obstacle Center Right Sensor	0 – 255
Obstacle Side Right	Obstacle Side Right Sensor	0 – 255
Line Left	Line left Sensor	0 (white) – 255 (black)
Line Right	Line right Sensor	0 (white) – 255 (black)
Light	Ambient Light Sensor	0 – 100%
Distance	Km counter in mm	-
Microphone	Noise level	0 – 255
X-Axis g	Acceleration in g in the axis X	-2.0 to 2.0 g
Y-Axis g	Acceleration in g in the axis Y	-2.0 to 2.0 g
Z-Axis g	Acceleration in g in the axis Z	-2.0 to 2.0 g
Motor End	Activated sensor when the last command sent to the motor has finished.	0 – 1

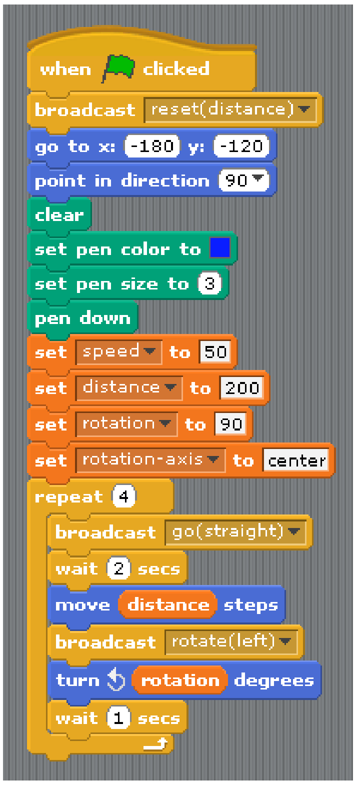
8.5. Exercises

For a better understanding of the mOway integration within the MowayWorld installer we include five sample exercises. By default they are located in this folder: “*Program Files -> mOwayPack v3 -> mOwayRobot -> mOway Scratch projects*”. You can find them also in “*Start -> All programs -> mOway Pack v3 -> mOway Scratch projects*”.

8.5.1. Geometric shapes

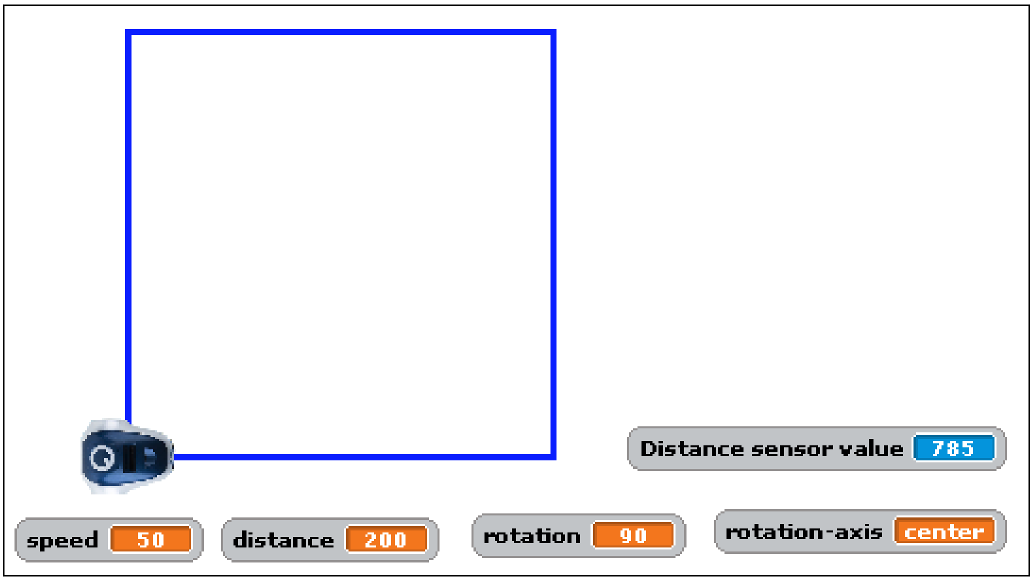
The following Scratch program reproduces geometric shapes. The examples can be found in the Scratch projects folder in mOway Pack, named “moway_square.sb” and “moway_traingle.sb”.

The geometric shape is achieved by varying the number or iterations (the polygon sides), rotation (mOway rotation angles) and “distance” (side length). mOway reproduces the same geometric figure than Scratch in the real world.



```
when ⚑ clicked
broadcast reset(distance) ▾
go to x: -180 y: -120
point in direction 90 ▾
clear
set pen color to ■
set pen size to 3
pen down
set speed ▾ to 50
set distance ▾ to 200
set rotation ▾ to 90
set rotation-axis ▾ to center
repeat 4
    broadcast go(straight) ▾
    wait 2 secs
    move distance steps
    broadcast rotate(left) ▾
    turn ↺ rotation degrees
    wait 1 secs
```

The result is the following: it is observed that the mOway odometer marks about (with a margin of error) 200 mm x 4 sides covered by mOway.



Distance sensor value 785

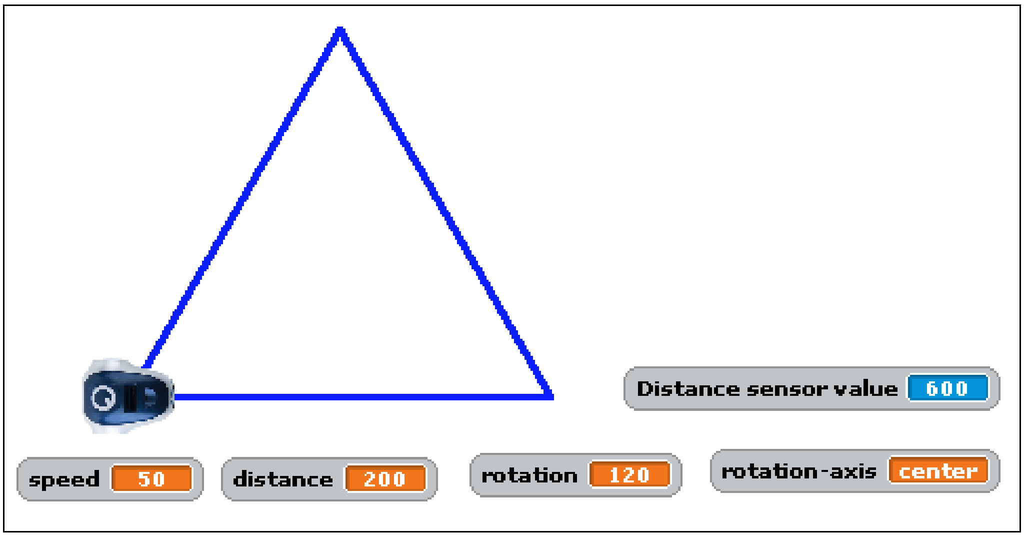
speed 50

distance 200

rotation 90

rotation-axis center

If the variable “rotation” is changed to 120º and the number of iterations of the “repeat” block is changed to 3, the result will be an equilateral triangle.



Distance sensor value 600

speed 50

distance 200

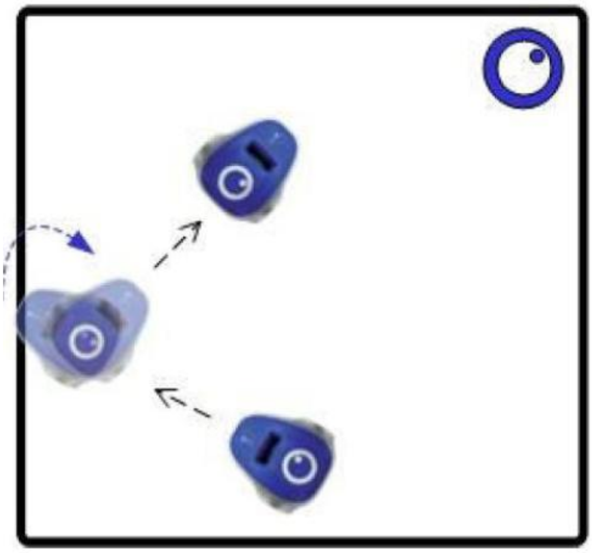
rotation 120

rotation-axis center

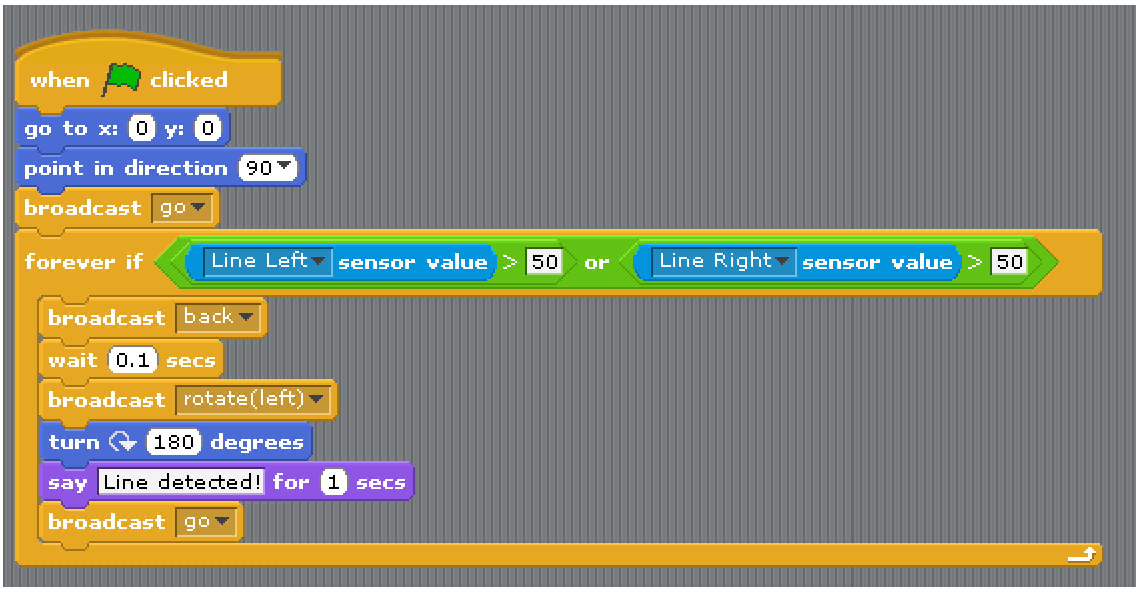
8.5.2. Enclosed

The purpose of this exercise is that mOway move forward until it reaches a black line. This program can be found in the Scratch projects folder in mOway Pack, named “moway_enclosed.sb”

When one of the two line sensors detects a black line, mOway goes back a bit, turn around and continue. Thus, mOway will be enclosed within a black contour on a white surface.



At the same time the detection will be also reflected in Scratch. Each time mOway detects a line the object will rotate and it will alert that a line is detected.

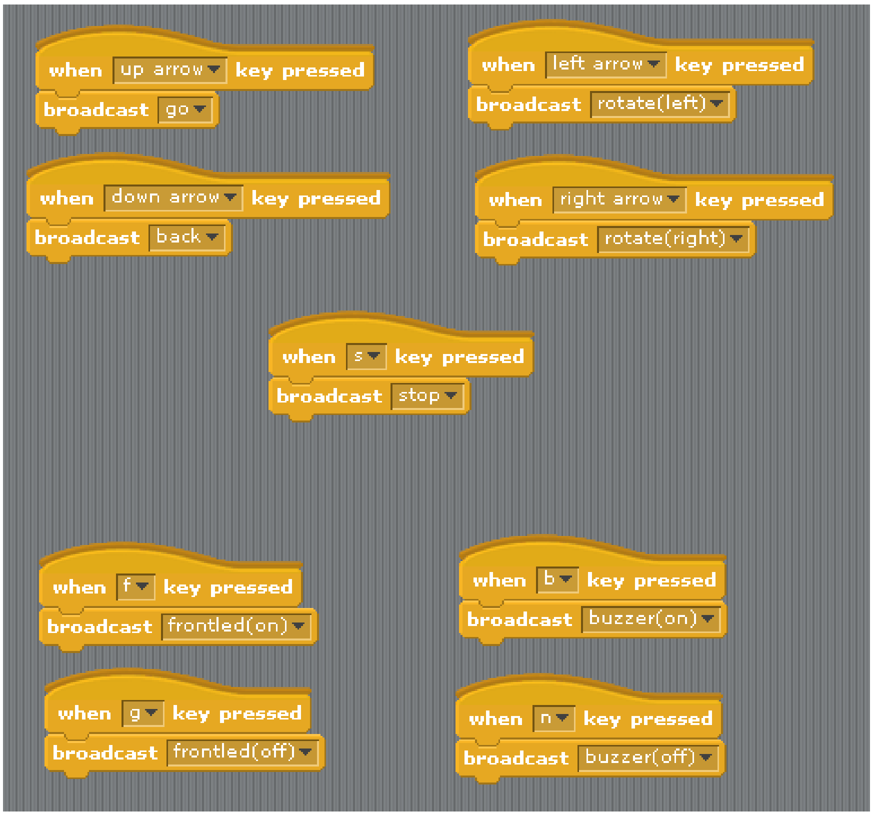


```
when [flag] clicked
go to x: 0 y: 0
point in direction 90
broadcast go
forever if < Line Left sensor value > 50 or Line Right sensor value > 50 >
    broadcast back
    wait 0.1 secs
    broadcast rotate(left)
    turn 180 degrees
    say Line detected! for 1 secs
    broadcast go
```

8.5.3. MowayRC

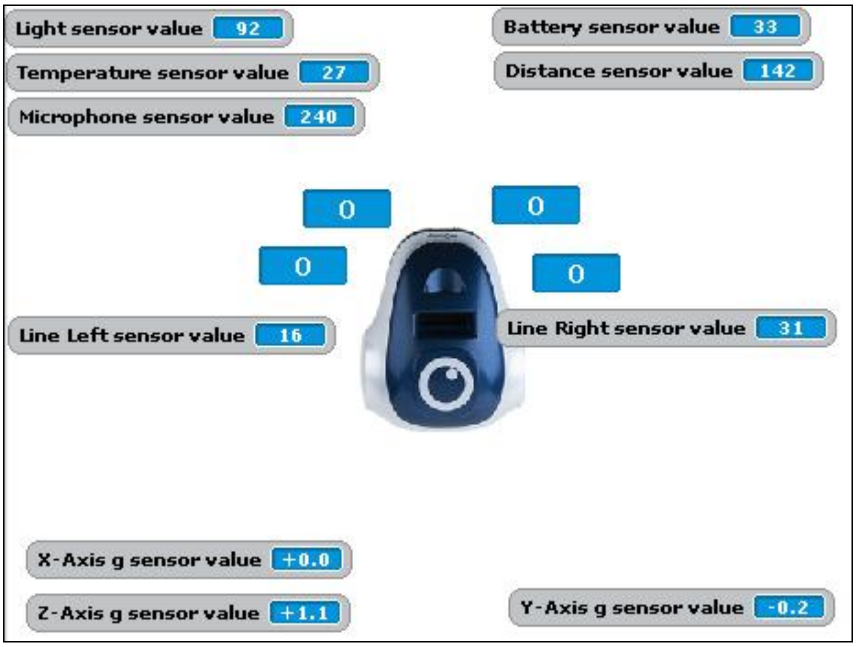
The following program acts as a mOway radio-control using Scratch. This program can be found in the Scratch projects folder in mOway Pack, named "moway_RC.sb"

With the arrow keys we control the mOway movement and with some additional keys we activate the frontal LED and the buzzer.



```
when up arrow key pressed
broadcast go

when left arrow key pressed
broadcast rotate(left)

when down arrow key pressed
broadcast back

when right arrow key pressed
broadcast rotate(right)

when s key pressed
broadcast stop

when f key pressed
broadcast frontled(on)

when b key pressed
broadcast buzzer(on)

when g key pressed
broadcast frontled(off)

when n key pressed
broadcast buzzer(off)
```

In this scene we observe the mOway sensors value in real time.



Light sensor value 92

Battery sensor value 33

Temperature sensor value 27

Distance sensor value 142

Microphone sensor value 240

Line Left sensor value 16

Line Right sensor value 31

X-Axis g sensor value +0.0

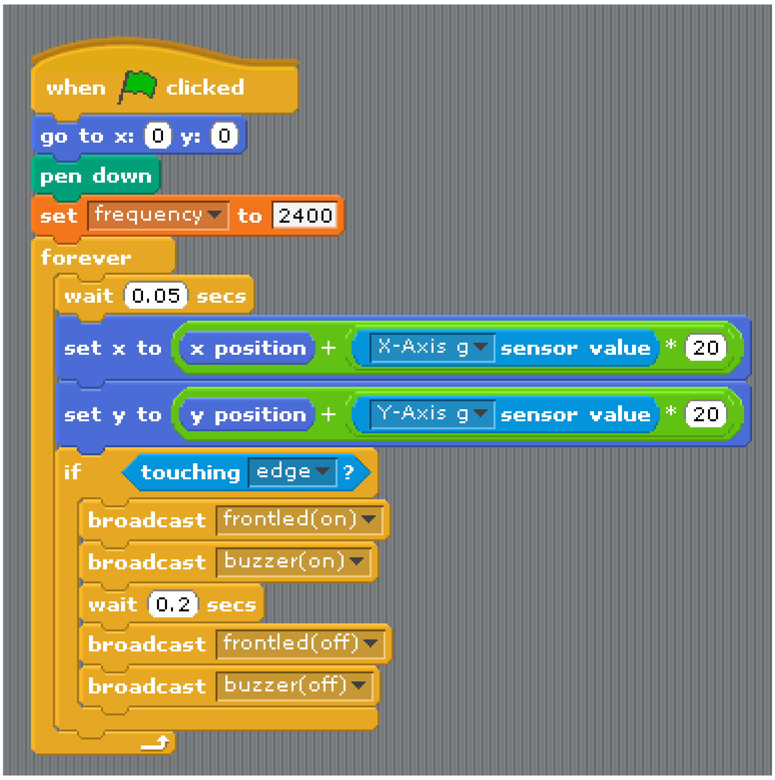
Z-Axis g sensor value +1.1

Y-Axis g sensor value -0.2

8.5.4. *Accelerometer*

In this exercise mOway acts as a remote control of a console game using the accelerometer. This program can be found in the Scratch projects folder in mOway Pack, named “moway_accel.sb”

The purpose is to keep mOway inside the square. Tilting mOway, the Scratch objet will move across the screen. In the case we touch the edge mOway will activate the buzzer and the front LED.



```
when ⚑ clicked
go to x: 0 y: 0
pen down
set frequency to 2400
forever
    wait 0.05 secs
    set x to (x position + (X-Axis g sensor value * 20))
    set y to (y position + (Y-Axis g sensor value * 20))
    if touching edge ?
        broadcast frontled(on)
        broadcast buzzer(on)
        wait 0.2 secs
        broadcast frontled(off)
        broadcast buzzer(off)
```

We can change the exercise adding more obstacles and more difficulty.

mOway
MILAN
430
STAEDTLER
MADE IN GERMANY