MOWAY'S
BEGINNER MANUAL

Copyright (c) 2010 Bizintek Innova, S.L.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 2.0 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

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.

- An easy and entertaining way to learn.
- This book's purpose: to be mOway's Manual and not a comprehensive book on minirobotics.
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 allows it to move 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
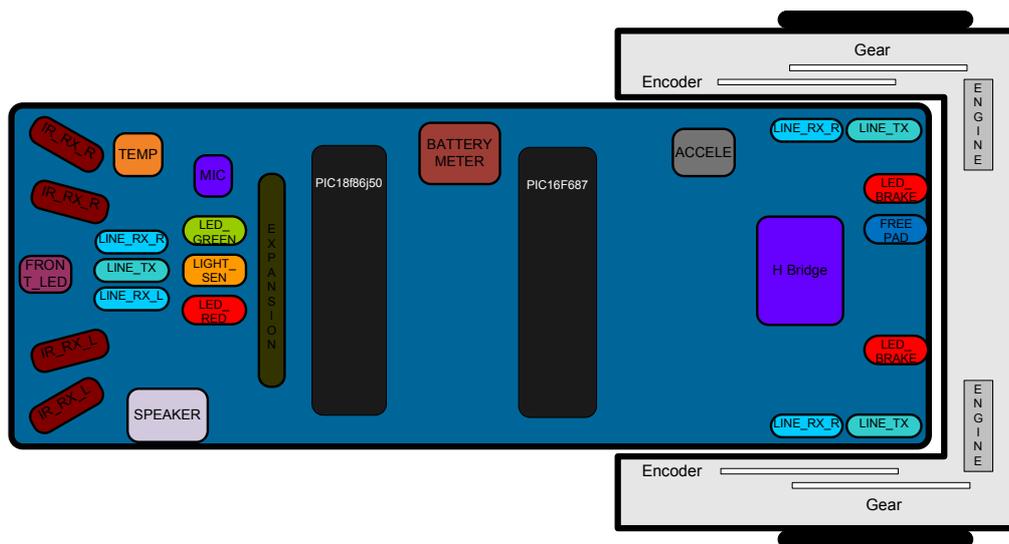
Image 1. Diagram of mOway's parts

3.1.   Processor

mOways are governed by a 4 Mhz PIC18F87J50 microcontroller manufactured by Microchip Technologies. All 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.
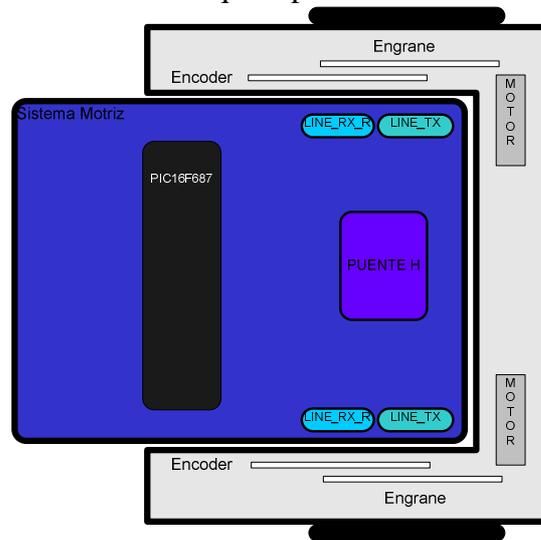
Image 2. Drive system: electronic and mechanical

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. **Traveled distance control**: Controls the distance traveled by each command with a precision of 1 mm aprox.
4. **General speedometer**: counts distances traveled 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.
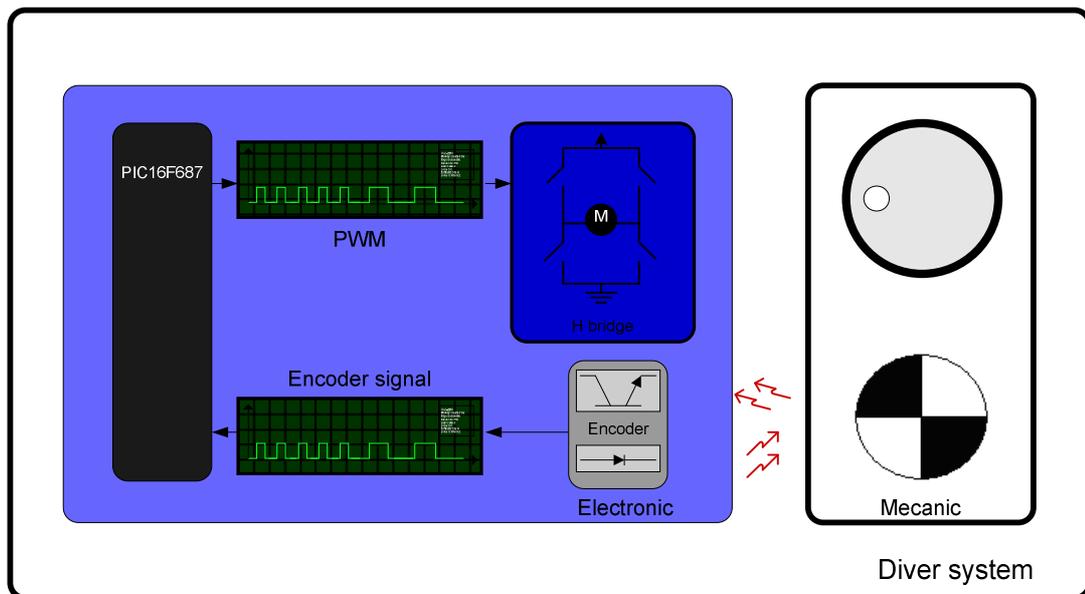
Image 3. Motor control

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.
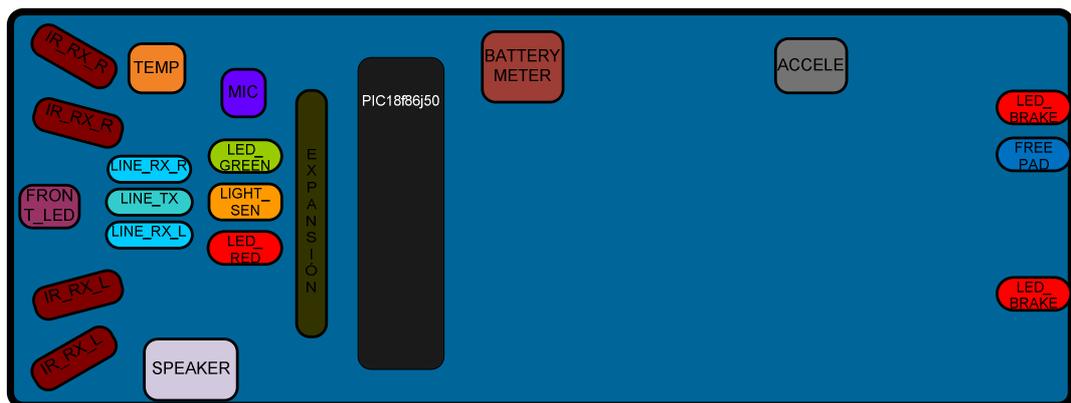
Image 4. Sensors and indicators group

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 color 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.
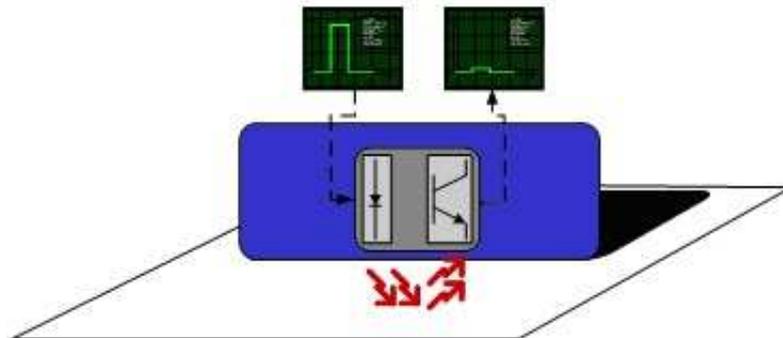
Image 5. Line tracking sensor on a clear surface.

- **Colored surface**: A colored surface reflects part of the emitted light obtaining an intermediate voltage at the microcontroller's analog channel input. This way colors are easily identified[1].
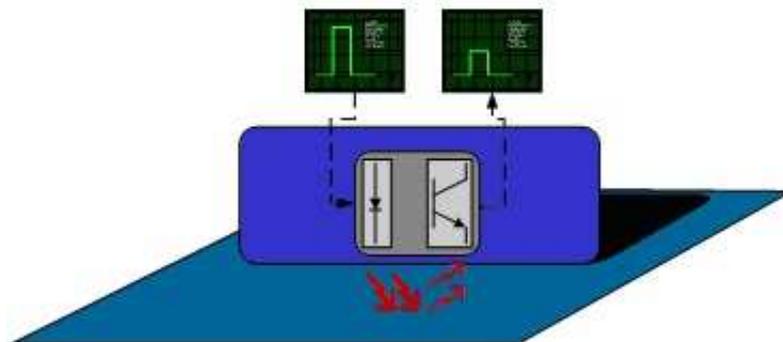
Image 6. Line tracking sensor on a colored surface.

1. **Dark surface**: A dark surface reflects very little light obtaining a high voltage reading at the sensor's output.
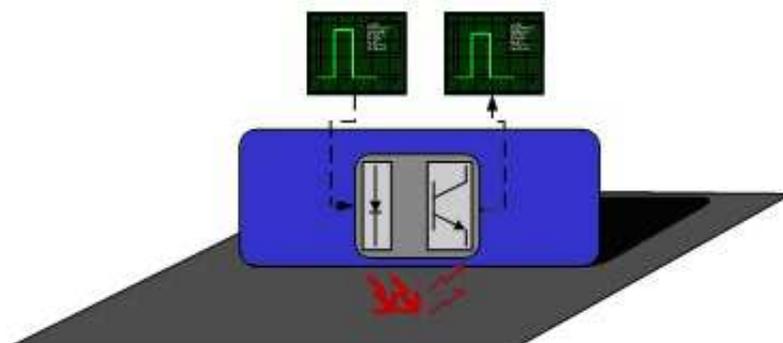
Image 7. Line tracking sensor on a dark surface.

---

[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 allows 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.
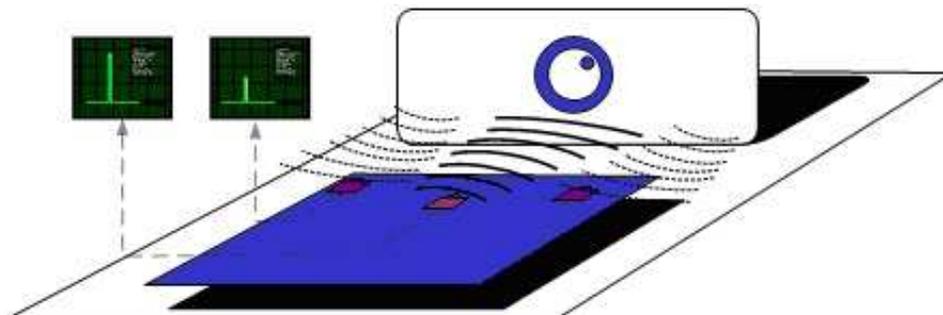
Image 9. Obstacle detection sensor

---

[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 allows 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-color 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*[9]

RF module allows communicate with 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[11].

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

---

[11] Option not yet available

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.   *Camera module and mOway Videocap*

Thanks to the camera module (**mOway Camera Board**[12]) it is possible to display on the computer what mOway is "watching" . Camera board sends images wirelessly to the video **mOway Videocap**[13].

Image 18. Camera module mOway Camera Board

Camera module is connected in expansion connector and it is very easy to use with mOwayWorld[14]. Video **mOway Videocap** must be connected to an USB connector of the computer.

Camera control is performed by Microchip **MCP23S08 device**, which is an input/output port managed by SPI. A library is provided to make easier for user to develop programs with the camera module. The basic functions are the camera ON and OFF, and the selection of the transmission channel from de camera to the video **mOway Videocap**. User have to select the same channel (from 1 to 4) both in program and **mOway Videocap**. The status of the camera is shown by a LED on camera board.

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.

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

---

[12] Available in some packs
[13] Available in some packs
[14] Option not yet available

4. First Steps

4.1.   mOway Pack installation

In mOwayPack (available in the webpage or in the installation CD) you will find the software, mOway´s libraries, test programs and documentations.

Following setup steps you will have all the resources:

- Beginner's and User manual.
    - o Beginner's manual includes all you need to start working with mOway.
    - o User manual contains detailed description of the robot.
- mOwayWorld software.
    - o This software controls all aspects of the robot: program download, battery charge control, radio control[15], RFUsb[16] management and C or assembler programs download.
- Reference projects in assembler, C and mOwayWorld.
    - o Example projects to start working with mOway easily.
- RFUsb Driver
    - o Driver for RFUsb[17] that allows the communication between robots and PC.
- mOway Videocap Driver
    - o Driver for mOway Videocap [18] that allows to grab images from mOway camera to display on PC.

If a security warning message appears during installation, please click on "install driver anyway".

---

[15] Option not yet available
[16] Module not available in all kits
[17] Module not available in all kits
[18] Module not available in all kits
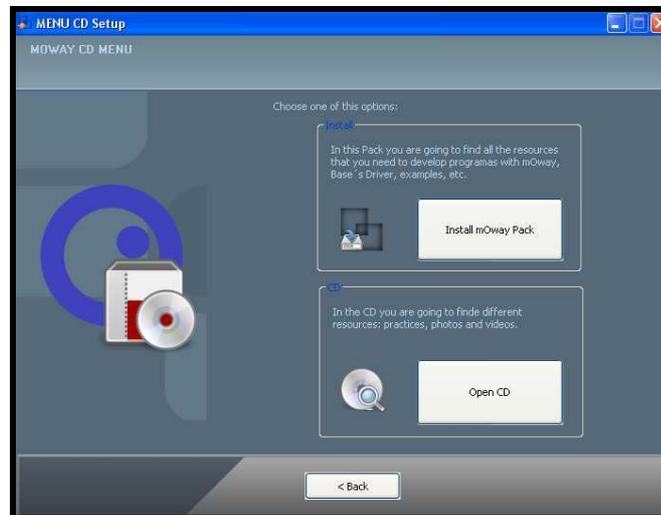
Image 19. CD

4.2. Download a program to mOway

Download process is always executed in mOwayWorld. This application can download to the robot mOwayWorld projects, assembler projects (compiled with Mplab o 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[19] a .HEX file from assembler or C project.
- Click download bottom. If a .HEX file has been imported the download progress will start automatically.
- Disconnect the robot and check the project.

mOwayPack provide 8 compiled projects: 3 to check sensors, 3 to check the drive system and 2 programs that are explained in the next section.

4.3. RFUsb instalation

- This is a device that allows to communicate the PC and mOway.

A driver that it´s 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.

---

[19] Option not yet available

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

Image 20. Driver installation Wizard

- Now the installation process will begin.
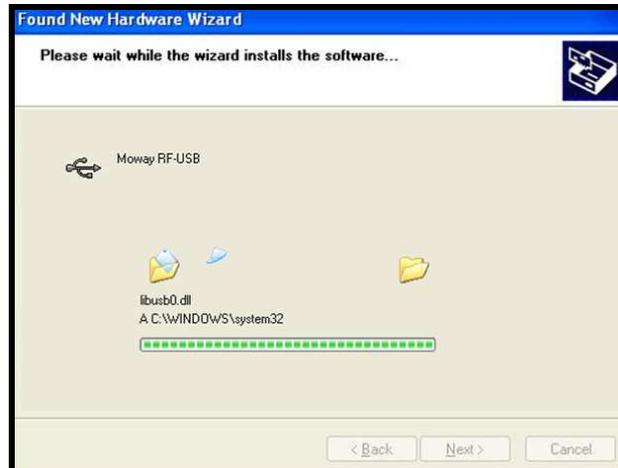
Image 21. Windows XP driver installation

- Assistant will then indicate that the hardware is installed.

Image 22. Driver installed in Windows XP

- Check if mOway's software has detected the RFUsb.

4.5   mOway Videocap drivers installation

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

- The first time the **mOway Videocap** 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 Videocap.**

5. mOwayWorld programming

mOwayWorld application allows 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.   Creating a Project

The first step to start to use mOwayWorld is to create a new project, by clicking on the "New project" icon. A name for the project and a path to save it must be chosen. The option "Activity Diagram (Graphical)" must be selected.
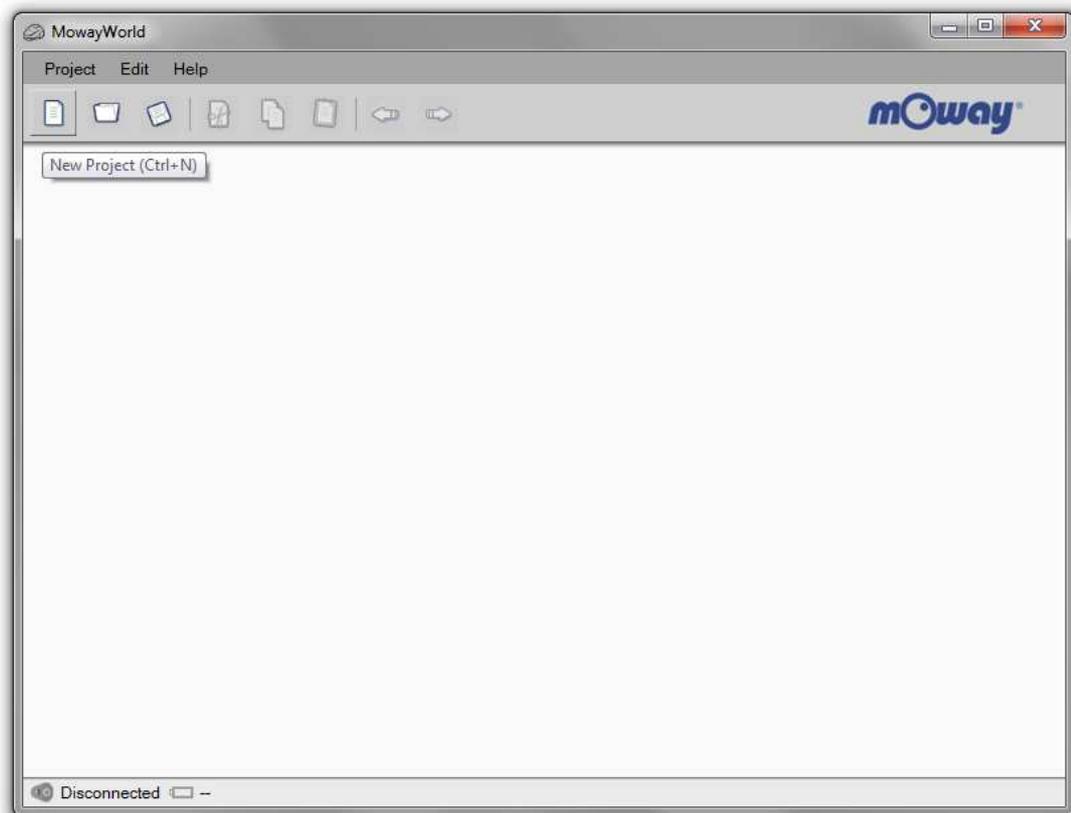
Image 23. mOwayWorld initial screen

5.2. mOwayWorld workspace

The following lines describe the different parts of mOwayWorld workspace.

5.2.1. Toolbar

Toolbar allows the user to manage the project, edit the flowchart, create variables, program mOway robot and many other functions.

Image 24. Toolbar

5.2.2. 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.
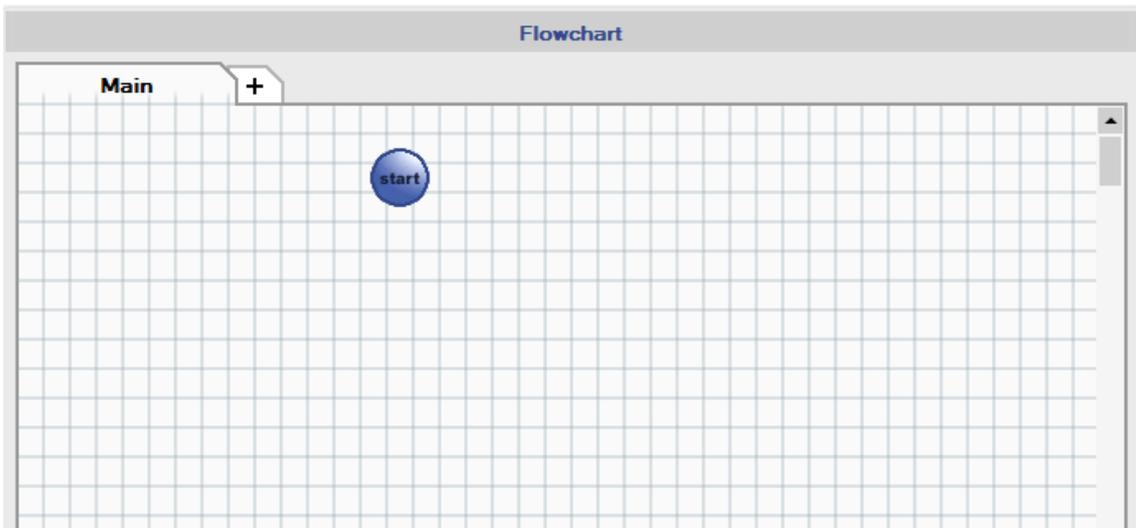
Image 25. Flowchart Editor window

5.2.3. Toolbox

On the left side of the workspace we can find the Toolbox 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).
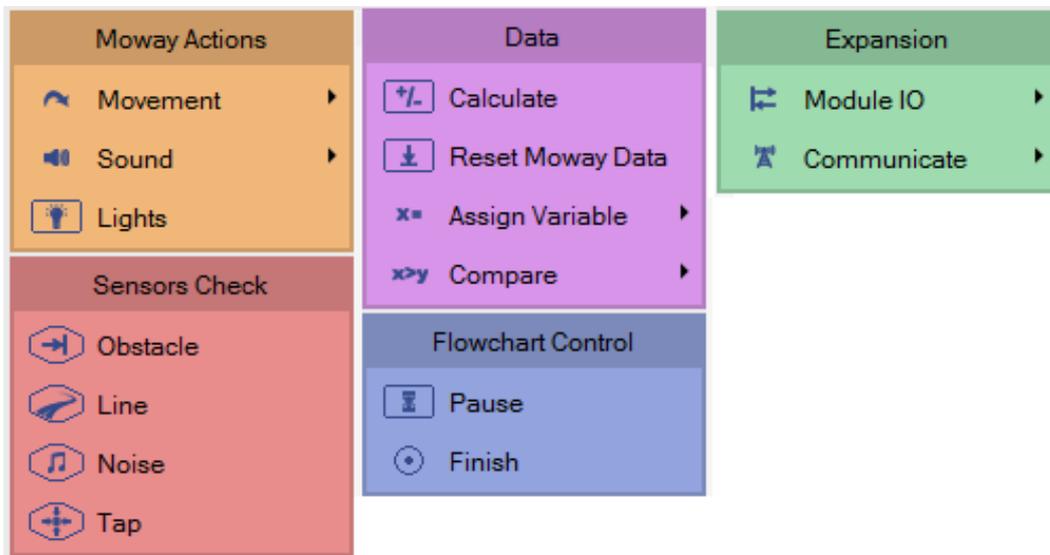
Image 26. Toolbox section

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. Some modules are grouped by a more specific type (Movement, Sound,…). This is represented by a little black arrow in the toolbox module types.

5.2.4.   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.
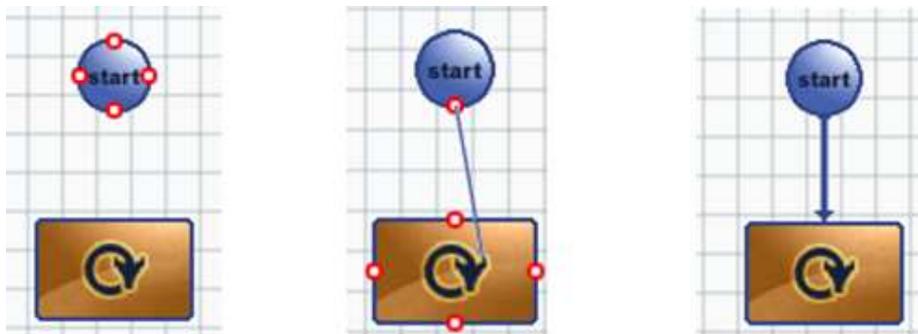
Image 27. Arrow drawing

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.
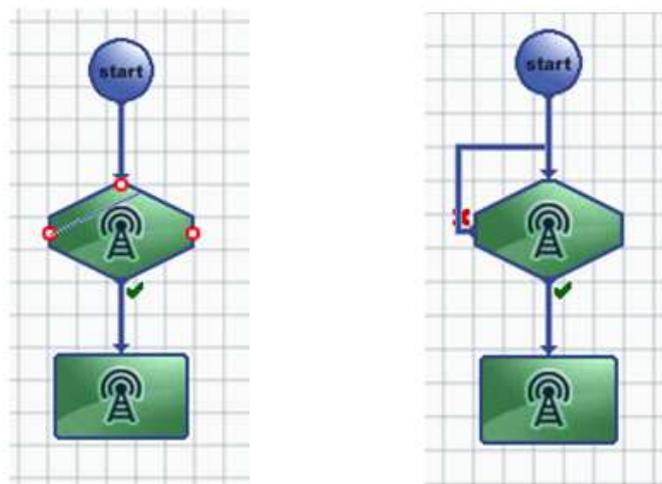
Image 28. Arrow drawing for a loop

If more precision is needed, the connector tool is very useful. To activate it, click on "Connect tool" icon in the toolbar.

5.3.   *First program in mOwayWorld*

In order to develop your first program, first you must create a project (see previous chapter). This first basic program will make mOway avoid obstacles.

1. Once a new project is created, the program starts with a 2-second delay. Just add a "Pause" module and configure it with a constant value of 2.

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

Image 29. 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 lights up.
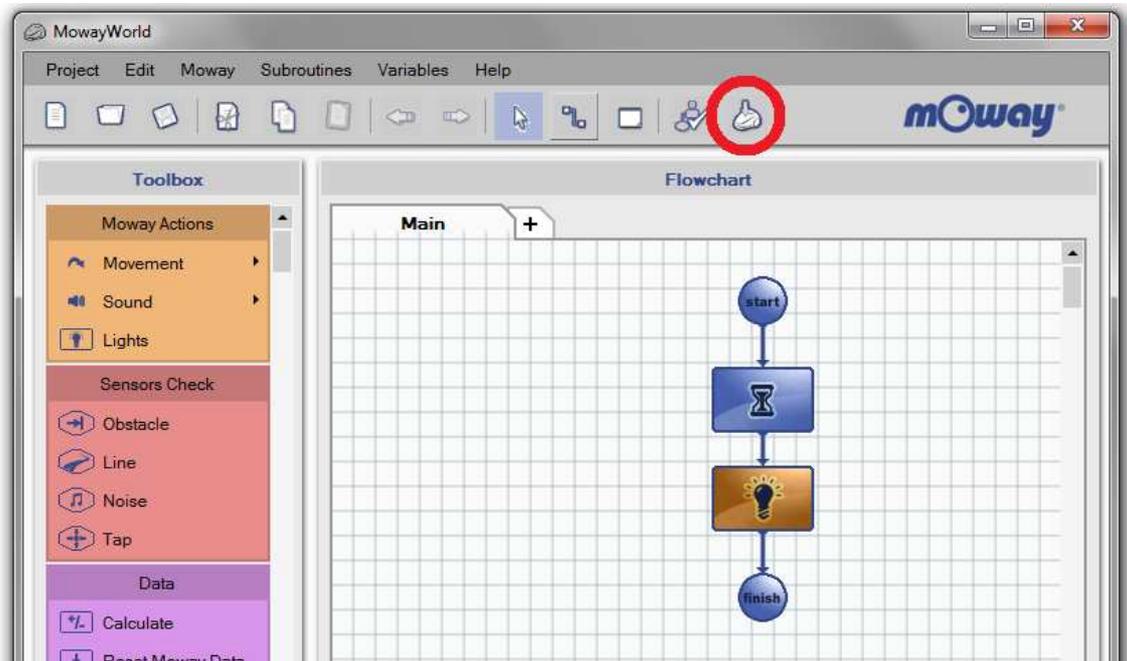
Image 30. Flowchart of the program and "Program mOway" button

6. In order to detect obstacles, four "Sensor Check Obstacle" has to be added and configured one for each obstacle sensor.
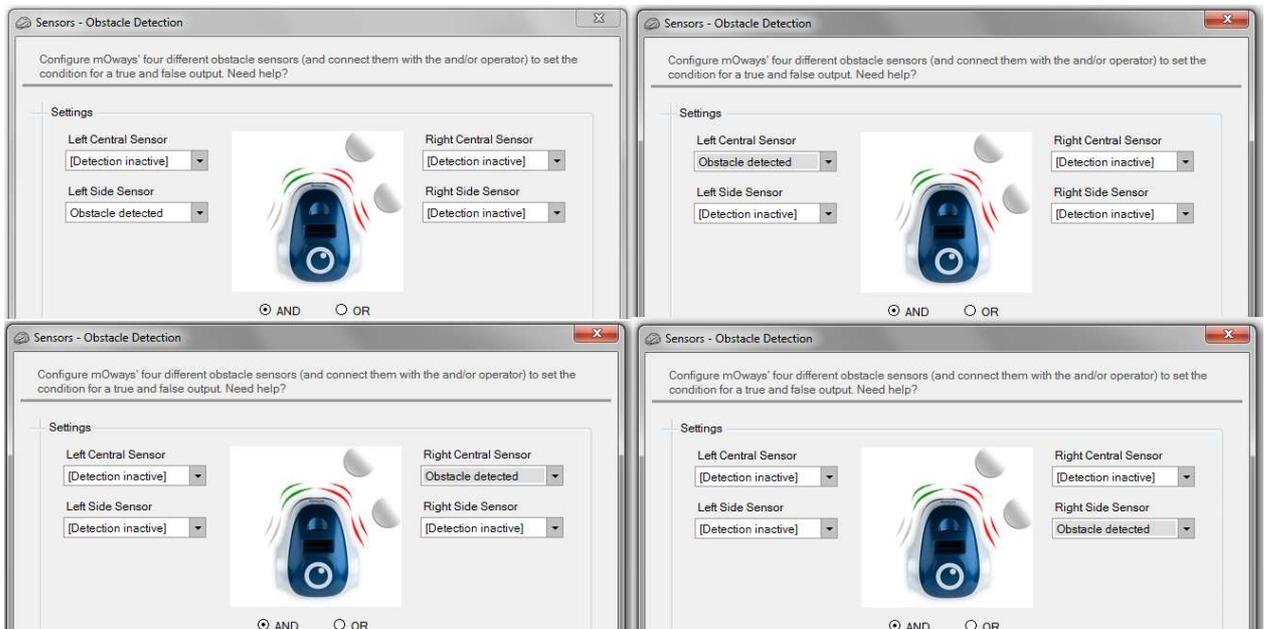
Image 31. Check Obstacles configuration

7. Condition modules have a true output and false output. If the condition is true (obstacle detected) the corresponding LED lights up, otherwise it remains off.

8. Test the program and check that the front LEDs light up when an obstacle is detected.
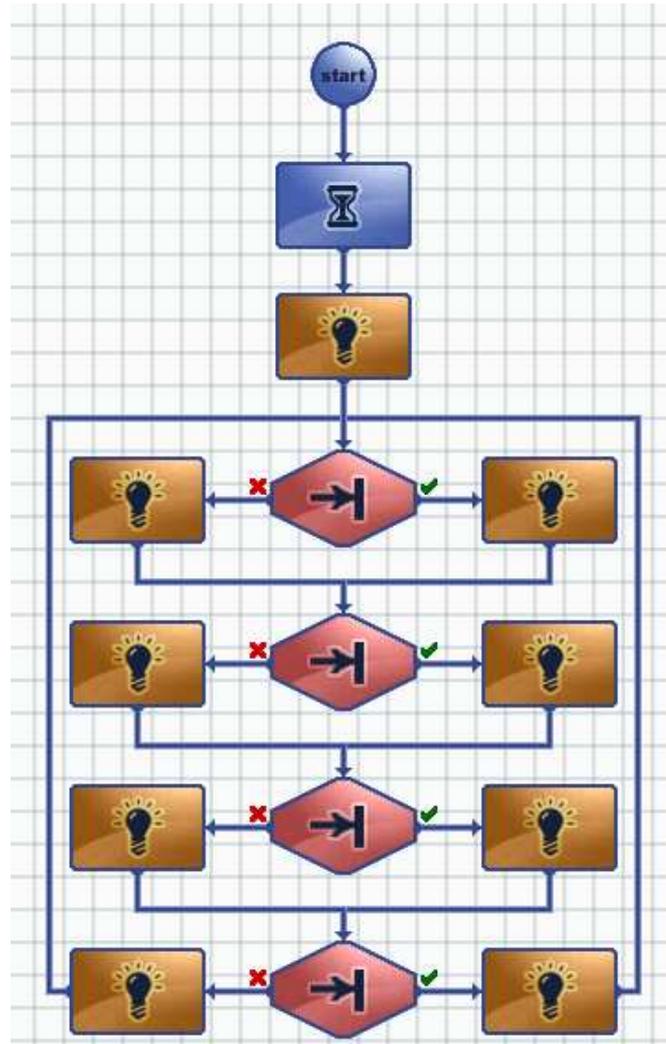
Image 32. First mOwayWorld program: obstacle detection

9. We add movement to the robot: straight on 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.
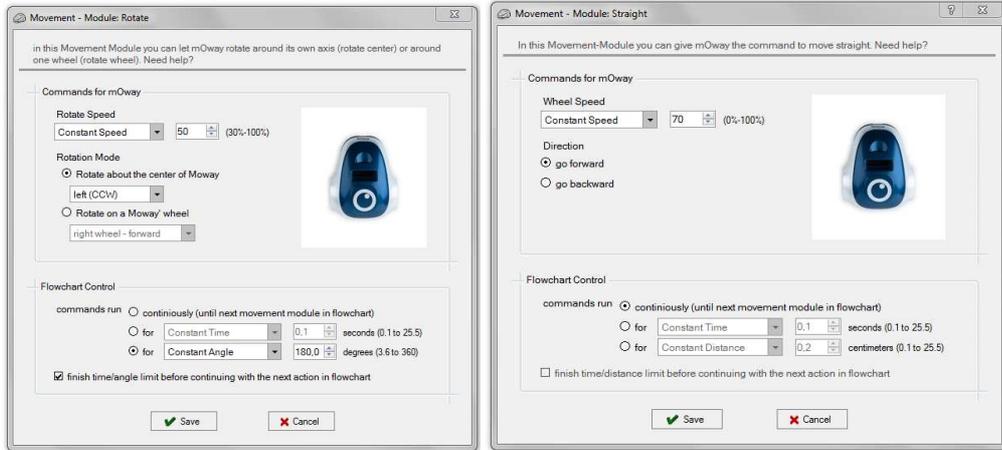
Image 33. Movement and Rotation configuration
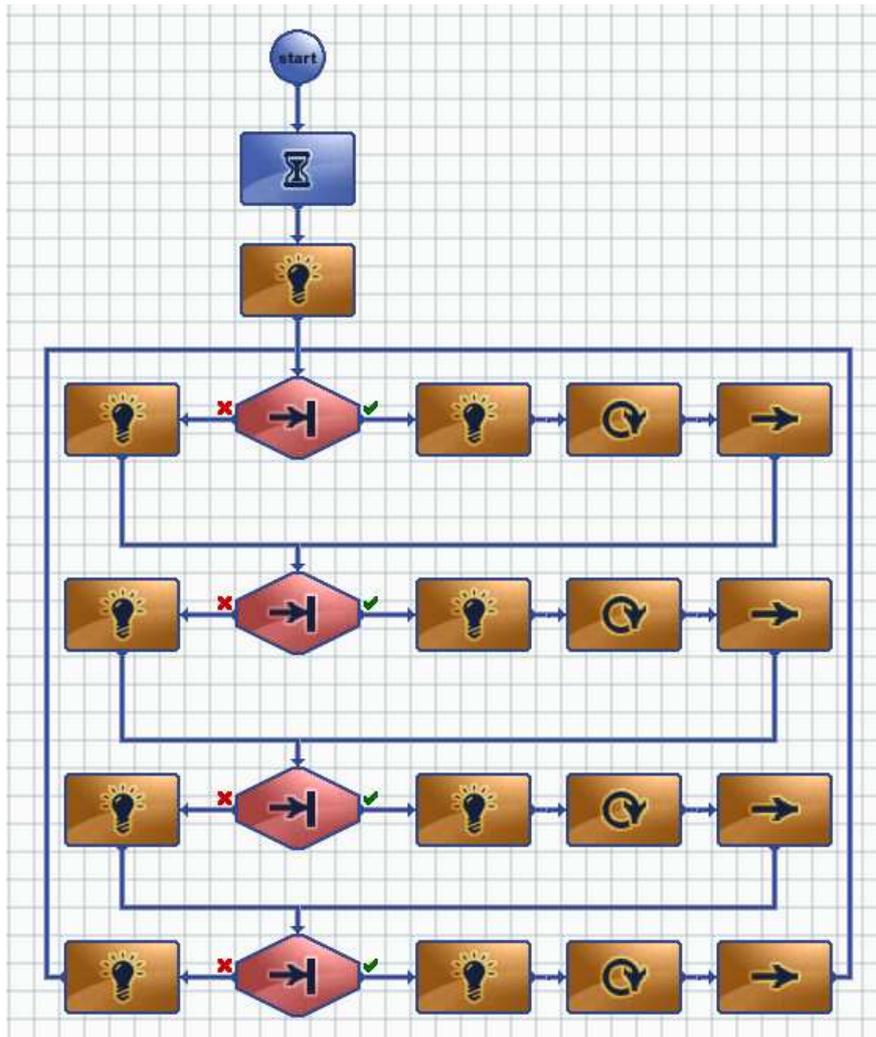
Image 34. End of first program in mOwayWorld

5.4.   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.

For more flexible programs some modules allows to use variables for some of their values (for example, speed value, pause time, read sensor values, etc.). A variable contains a value that can change during the execution of the program. These 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 selected.
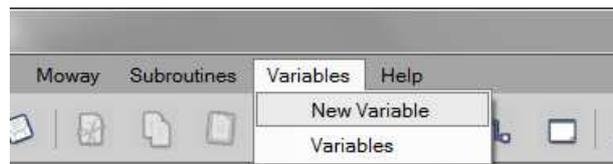
Image 35. A new variable creation.

Each of the modules included in mOwayWorld are described below.

5.4.1.   *Moway Actions*

This group of modules allows to control mOway's actuators: engines, 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 are 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.
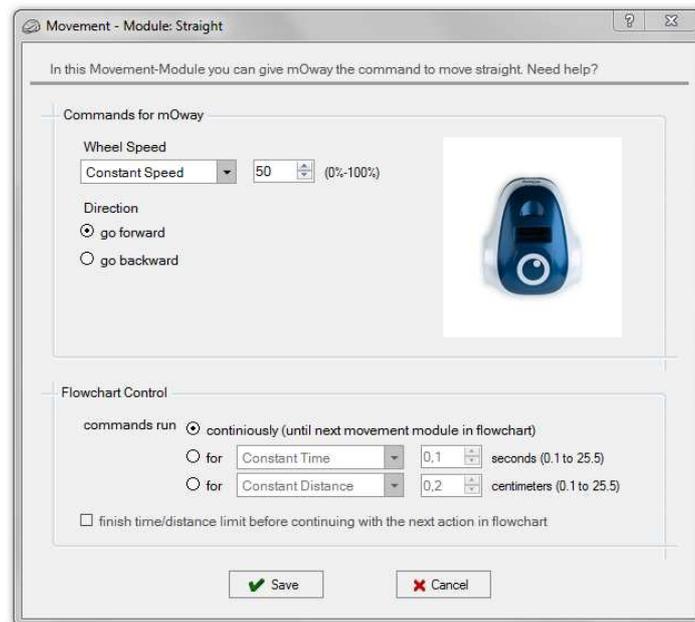
Image 36. Movement – Straight configuration window

- **Movement – Free**

Free movement is similar to straight movement, but in this case the speed of each motor can be adjusted separately.
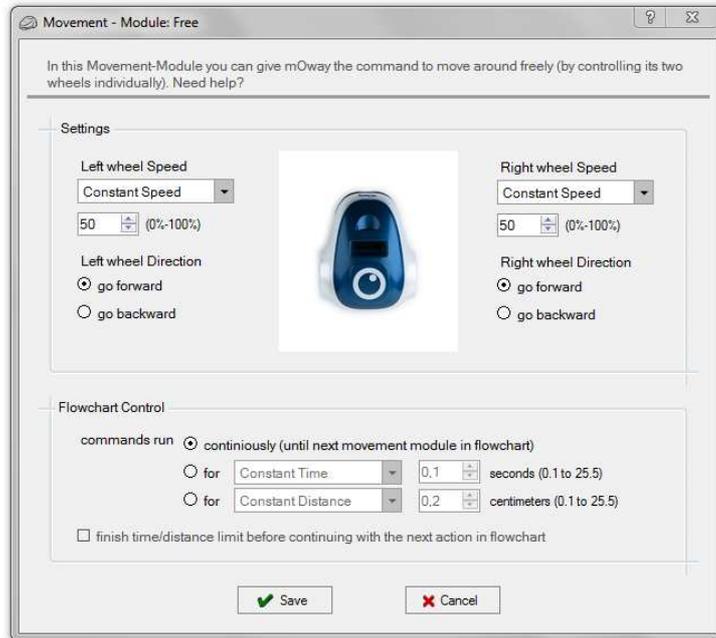
Image 37. Movement – Free configuration window

- **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 radius.
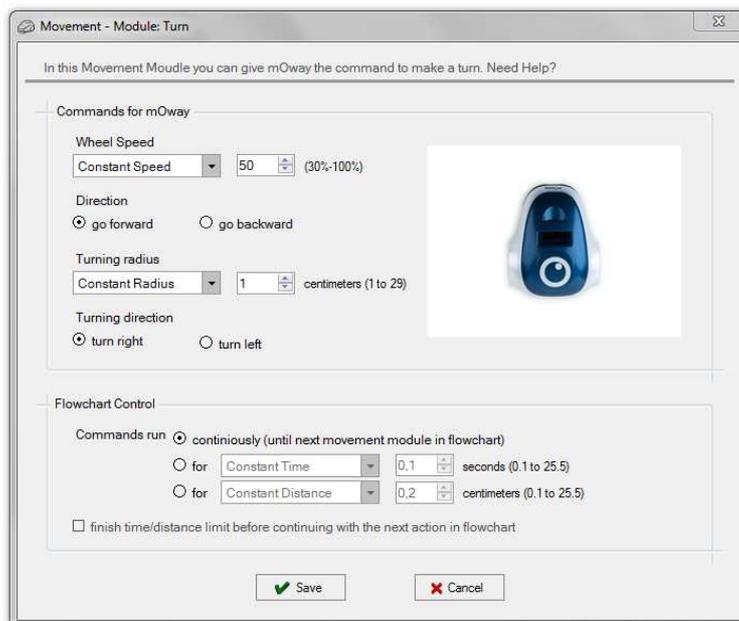
Image 38. Movement – Turn configuration window

- **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.
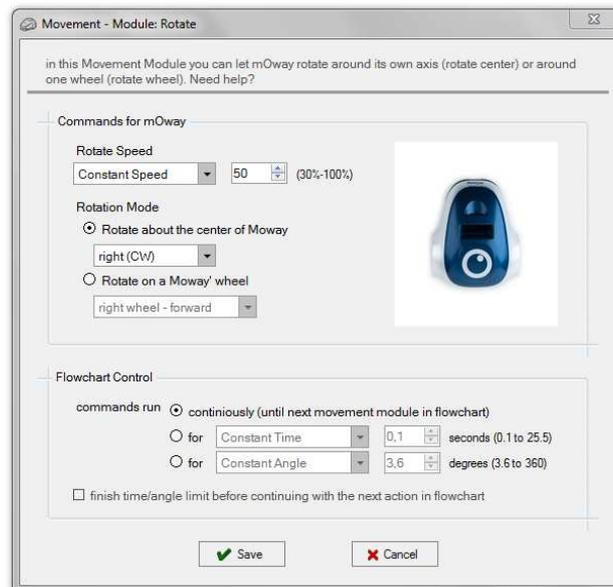
Image 39. Movement – Rotate configuration window

- **Movement – Stop**

This command stops mOway's engines.

- **Sound – Play**

This function allows mOway to emit tones from 250 Hz to 65 KHz. It is possible to define the time that the speaker will be activated.
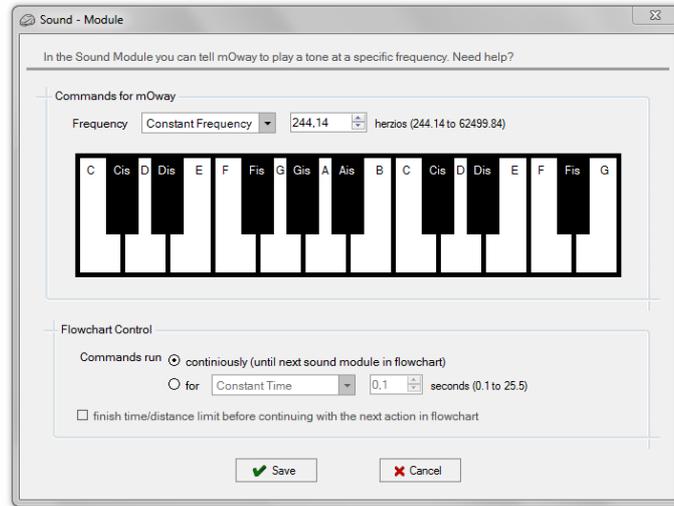
Image 40. Sound – Play configuration window

- **Sound – Stop**

Stops the speaker sound.

- **Lights**

This module allows to operate on mOway's LED diodes. You can turn them on, turn them off or make them blink.
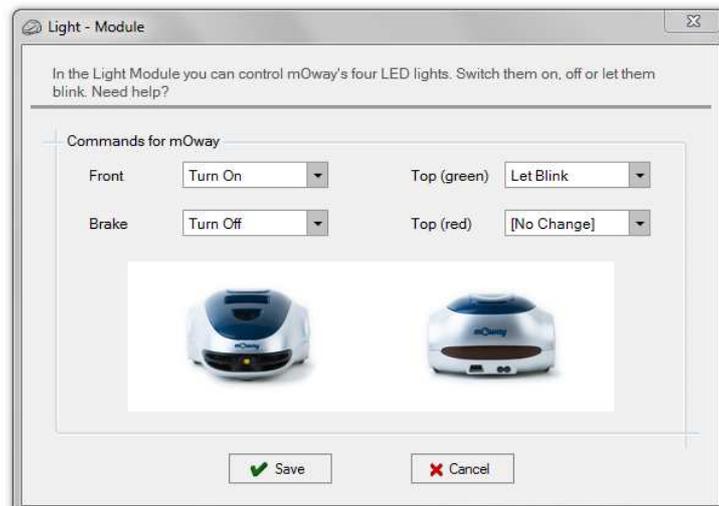
Image 41. Lights configuration window

5.4.2.   Sensors Check

This group of modules allows 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 in the flowchart). Otherwise, if the condition is false, the output will be "false" (red cross in the flowchart).

- **Obstacle**

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

- o obstacle detected
- o no obstacle detected
- o detection inactive
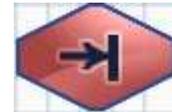
In addition, it allows to check the AND or OR boolean operation. If AND option is checked, all the conditions must be true to get a "true" output. On the other hand, if OR is checked, just one of all the condition has to be true to get a "true" output.
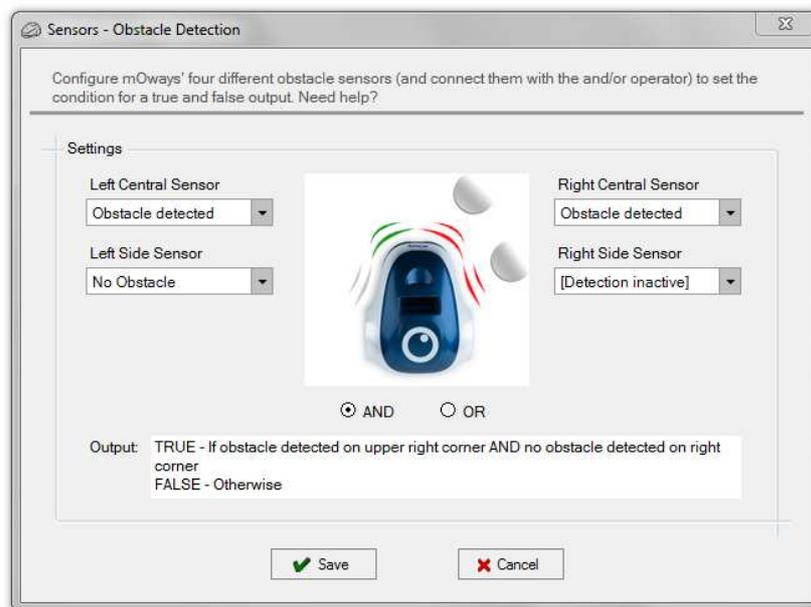
Image 42. Sensors Check - Obstacles configuration window

- **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:

- o black line detection
- o white line detection
- o detection inactive

In addition, it allows 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.
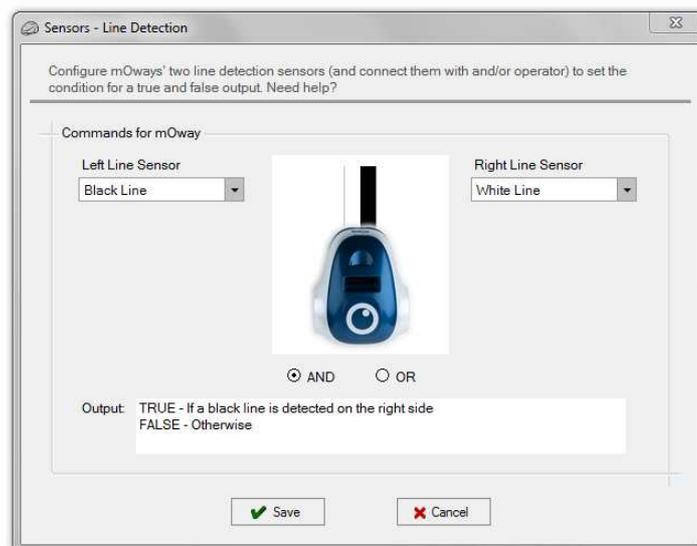
Image 43. Sensors Check – Line configuration window

- **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.

5.4.3.   *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 previous 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 goes 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 allows to assign a value to a user-defined variable. This value can be constant or the analog value of mOway's sensors.

  - o **Value:** assigns a constant value to a variable.

  - o **Time:** assigns elapsed time to a variable. Value goes from 0 to 255 tenths of a second.

  - o **Speed:** assigns the speed value of one of the wheels to a variable. Value goes from 0 (stopped) to 255 (maximum speed).

o **Distance:** assigns distance value covered by mOway to a variable. Value goes from 0 to 255 centimetres.

o **Angle:** assigns the value of mOway turning angle to a variable. Value goes from 0 (0 degrees) to 100 (corresponding to 360 degrees).

o **Brightness:** assigns the light sensor value to a variable. Value goes from 0 (dark) to 100 (light).

o **Line:** assigns one of the line sensors value to a variable. Value goes from 0 (white colour detection) to 255 (black colour detection).

o **Obstacle:** assigns one of the obstacle sensors value to a variable. Value goes from 0 (no detection) to 100 (closest detection).

o **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).

o **Noise:** assigns microphone value to a variable. Value goes from 0 (silence) to 255 (loud noise).

o **Temperature:** assigns the robot temperature value to a variable. Value goes from 0ºC to 255ºC.

o **Battery:** assigns battery level to a variable. Value goes from 0% to 100%.

- **Compare**

This group of modules allows 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 (<=).

o **Data:** compares a variable with a constant value or a user-variable.

o **Time:** compares elapsed time. Values goes from 0.1 to 25.5 seconds.

o **Speed:** compares one of the wheels speed value. Values goes from (stopped) to 100 (maximum speed).

o **Distance:** assigns distance value covered by mOway to a variable. Value goes from 0 to 255 centimetres.

o **Angle:** compares the value of mOway turning angle. Value goes from 0 (0 degrees) to 100 (corresponding to 360 degrees).

o **Brightness:** compares the light sensor value. Value goes from 0 (dark) to 100 (light).

o **Line:** assigns one of the line sensors value to a variable. Value goes from 0 (white colour detection) to 100 (black colour detection).

o **Obstacle:** compares one of the obstacle sensors value to a variable. Value goes from 0 (no detection) to 100 (closest detection).

o **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).

o **Noise:** compares microphone value. Value goes from 0 (silence) to 100 (loud noise).

o **Temperature:** compares the robot temperature value. Value goes from 0ºC to 255ºC.

o **Battery:** compares battery level. Value goes from 0% to 100%.

5.4.4.   *Flowchart Control*

- **Pause**

This module allows 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.

- **Finish**

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

5.4.5.   *Expansion*

This group of functions allows to control the expansion connector of mOway robot, either to use radiofrequency (RF) communication, or user expansion modules.

WARNING!
Only advanced users can use the pinout configuration. Any incorrect connection of electronic elements to the expansion connector may damage the robot irreversibly.

- **Module IO – Configure**

This module configures the expansion connector pins as inputs or outputs and assigns a initial value for outputs.
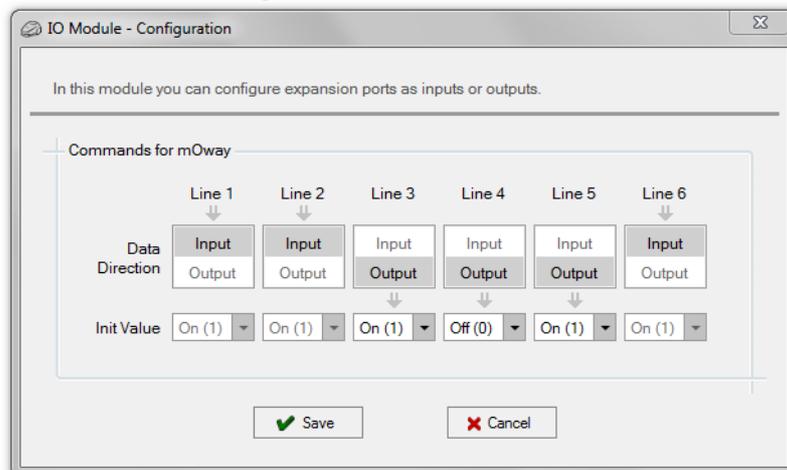
Image 44. Module IO – Configuration configuration window

- **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.
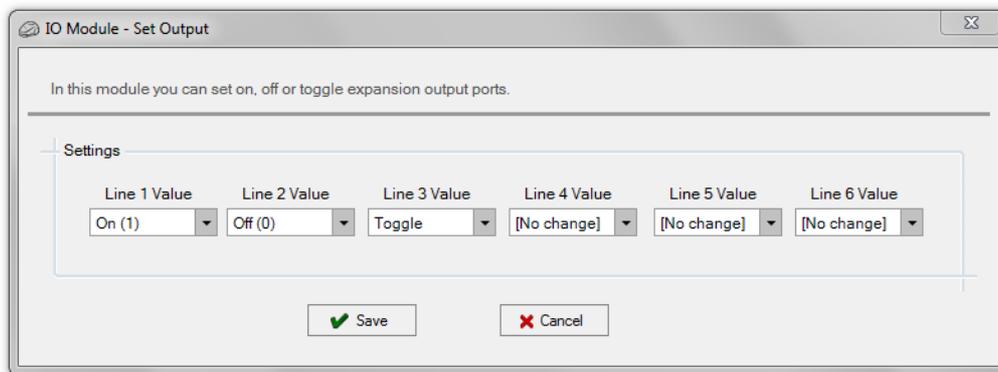
Image 45. Module IO – Set Output configuration window

- **Module IO – Check Input**

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

- **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 establish a communication between two mOways, they must be configured with the same channel and a different identification for each one.

If the configurations is correct, the output is "true". Otherwise, the output is "false".
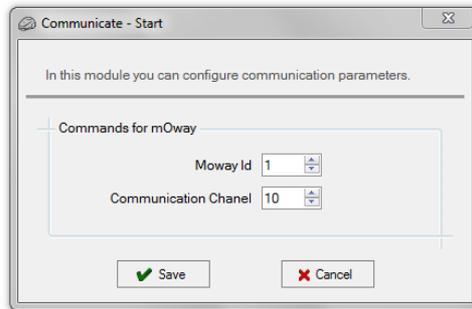
Image 46. Communicate – Start configuration window

- **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.
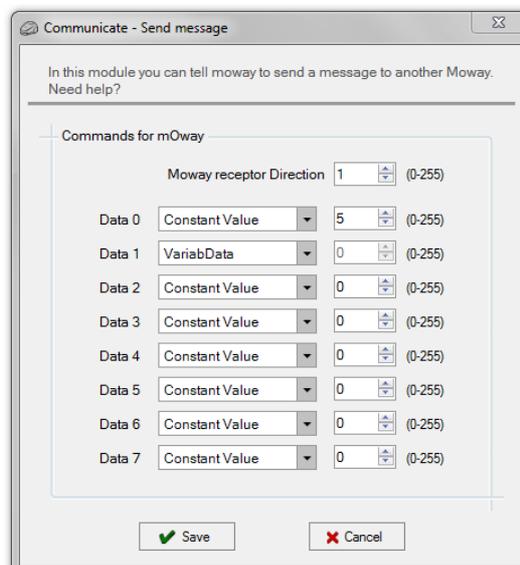
Image 47. Communicate – Send configuration window

- **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.
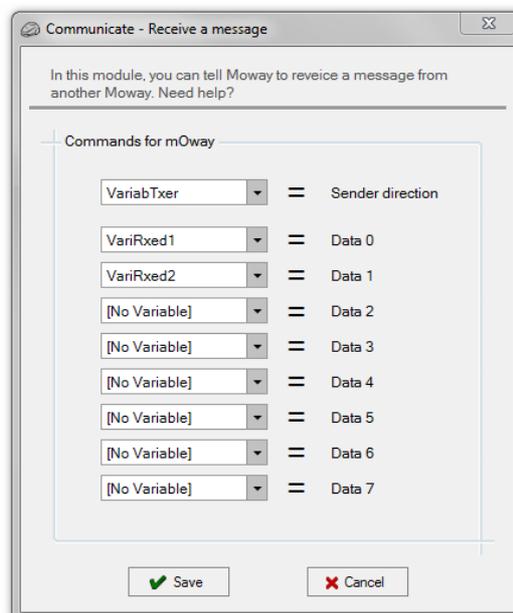
Image 48. Communicate – Receive configuration window