

BLOCKLY 1

Programming the Niryo One with Blockly



OBJECTIVES	3
REQUIREMENTS	3
HARDWARE INVOLVED	3
PROGRAMMING THE NIRYO ONE WITH THE LEARNING	3
PRESENTATION OF THE LEARNING MODE EXERCISE	3 4
PROGRAMMING THE NIRYO ONE WITH BLOCKLY PRESENTATION OF BLOCKLY THE PROGRAMMING ENVIRONMENT THE PROGRAMMING BLOCKS EXERCISE	5 5 7 8
APPENDIX	11
CORRECTION	16

OBJECTIVES

- Getting familiar with the basics of robotic programming
- Controlling the robot using Niryo One Studio and Blockly
- Being able to create algorithms to control the robot

REQUIREMENTS

Starter guide

WHAT YOU WILL NEED

- Computer equipped with WIFI
- Niryo One Studio (Niryo One's graphical programming interface)
- A Niryo One

PROGRAMMING THE NIRYO ONE WITH THE LEARNING MODE

PRESENTATION OF THE LEARNING MODE

The « **learning mode** » is a programming method in which the **human operator manually places a robot in a position** and then saves it to let the robot execute it later.

It is possible to program the Niryo One using this programming method. To enable or disable this mode, click the « **Learning mode** » button on the Niryo One Studio interface.

EXERCISE

- Activate the learning mode of the robot
- Move the robot with your hand to a position of your choice
- Hold the robot in this position and click on « **UPDATE VALUES** » before releasing the robot

You can also press the upper button on the robot to save the value

- Deactivate the « learning mode »
- Execute a movement to this position by clicking on « MOVE AXES »

Did the robot make the desired movement?

• We would like to put the robot in a « top » position as shown in the figure below:



Which axis will reach its maximum angle?

- Activate the « learning mode » and move the robot with your hand to this position
- Hold the robot in this position, select « UPDATE VALUES » and then save it with the name « TOP »
- Select this position and execute it by clicking on « MOVE AXES »

PROGRAMMING THE NIRYO ONE WITH BLOCKLY

In this section, we look at the basic functions of the visual programming software, based on Blockly.

PRESENTATION OF BLOCKLY



The visual programming interface of Niryo One Studio is based on Blockly, a Google library, which is also used for the Scratch project made by the MIT.



THE PROGRAMMING ENVIRONMENT

The different parts of the environment and their functionalities are :

Γ	והארכ	Robotics for Everyone	EN Connected to 10.1	0.10.10 (Niryo One Hotspot) 🗘 <
*	Logic Loops Math Lists			
+	Variables Functions			
13	Niryo One			
*				
Ā		Move Joints j1 (0) /2 #03521) /3 #01633 /4 (0.022) /5 (0.464) /6 #0.0153 Move Joints j1 (0.515) /2 #0.838) /3 #0.659 /4 (0.501 /5 (1.493) /6 #0.035		
inin SX		Move Joints (1 (1179) (2 (1032) (3 (10507) (4 (11805) (5 (11218) (6 (11463) Move Joints (1 (11172) (2 (10064) (3 (10550000000000000000000000000000000000		
~		Move Joints (1 11833) (2 10.02) (3 0.02) (4 12275) (5 11227) (6 11473) Move Joints (1 0) (2 0) (3 0) (4 0) (5 0) (6 0) Move Joints (1 0) (2 0) (3 0) (4 1220) (5 0003) (6 12222) Move Joints (1 0) (2 0) (3 10) (4 1220) (5 000300000000000000000000000000000000		
			earning Mode	STOP
			ints (rad) 0.000	Position (m) x:0.000
			0.000 0.000 0.000	y : 0.000 z : 0.000 Rotation (rad)
			0.000	x:0.000 y:0.000 z:0.000
		Arm max.	. speed - % 🖍	No tool selected
3				
C				

- $1 \rightarrow$ The workspace that will contain your code
 - **To delete a block, simply drag it to the trash can** You can also select it then press the « Del » key on your keypad
- 3 H Delete the current workspace
 - H Undo / Redo
- 5 \rightarrow Add a position block
- 6 H Execute the sequence displayed in the workspace
- 7 H Stop the execution of the current sequence in theworkspace
- 8 | Import or export a sequence
- 9 H The different programming blocks

THE PROGRAMMING BLOCKS

CONCEPT	EXPLICATION	EXEMPLE
Variables	Variable blocks are used to create variables and to use them in the program. A variable is a symbol that associates a name with a value.	set var v to (4 set var v to (var v
Lists	List blocks are used to store and then access a list of numbers and strings. A list is a structure that allows data to be grouped together in such a way that it can be accessed freely. It can be considered as an array.	create empty list
Loops	Loop blocks can be used for iteration (repetition of a series of instructions). A loop is a programming control structure that allows the repeated execution of a sequence of instructions.	repeat while v (true v) do Move Joints j1 -1.8 j2 0.1 j3 -1.2 j4 0 j5 0 j6 0 Wait for 3 seconds
Niryo One	Niryo One blocks are used to control the robot, its tools and its I/O interfaces.	Deactivate ▼ learning mode Set Arm max. speed to
Functions	Function blocks are used to create and call up functions and procedures. A function is a section that encloses a part of code performing a specific task that can be reused in the program.	Image: Constraint of the second state of the second sta
Logics	Logic blocks allow either to initiate an action according to a condition or to manipulate logic variables.	
Maths	Math blocks are used for mathematical operations.	set a v to random integer from v 1 to v 100 set a v to v 2 + v v b v

Refer to the appendix for more details on the different blocks.

EXERCISE

We want to move the Niryo One to the two positions P1 and P2 :

P1 = [x = -0.03 ; y = -0.156 ; z= 0.48 ; roll= -0.58 ; pitch = -0.58 ; yaw = -0.145]

P2 = [x = -0.136 ; y = -0.133 ; z = 0.255 ; roll= -0.081 ; pitch = 0.744 ; yaw = -2.535]

After reading the appendix, identify the block that allows you to move the robot to a position with Cartesian coodinate x y and z.

• Reproduce the following sequence in Niryo One Studio

Move Pose x -0.03 y -0.156 z 0.48 roll -0.58 pitch -0.58 yaw -0.145

Move Pose x -0.136 y -0.133 z 0.255 roll -0.081 pitch 0.744 yaw -2.532

- Deactivate the « learning mode » and execute the program
- Add a new block to your sequence as shown in the illustration below:



• Deactivate the « learning mode » and execute this new sequence

Explain the difference between the first and the second sequence and conclude on the role of the added block.

We want to automate the task of activating and deactivating the « learning mode ».

Which block should be used to perform this operation? Add this block at the beginning and at the end of your program, then execute it.

A sequence of instructions represents the program to be executed by the robot.

Keeping the program developed in the previous sections, click on the « import/export » button.



Name		
Descr	ption	
Pleakly		
<xml xi<br="">name=</xml>	mns="http://www.w3.org/1999/xhtml"> <block id="2H-t.aSy+N:*I`w+*a?r" type="niryo_one_activate_learning_mode" x="-770" y="-710"><field 'LEARNING_MODE_VALUE">0<next><block id="Q9PmGur00P6[Y =i%WSf" type="controls_repeat_ext"><value name="TIMES"><shadow< td=""><td></td></shadow<></value></block></next></field </block>	

A sequence is characterized by:

- Its name (by default, it will be « sequence » + ID)
- Its description



Knowing that:

- The positions of the robot are in joint coordinates
- The initial position of the robot is [0;0;0;0;0;0]
- P1 = [-1.215; -0.333; -0.867; -3.053; -0.014; 0.127]
- P2 = [-1.377; 0.22; -0.281; -3.053; -0.031; 1.549]

Translate the previous diagram into text.

• Reproduce the program below:

Deactivate v learning mode	
Move Joints j1 0 j2 0 j3 0	j4 0 j5 0 j6 0
Set Arm max. speed to 100	%
set counter to (0	

Choose the block that allows to make the loop of the program shown above.

Using the appendix, complete the Blockly sequence so that it reproduces the previous diagram.

Taking into account the counter variable and the loop block, propose a block to replace these two blocks.

- Replace this variable and loop blocks with this new block
- Execute the sequence

Comment.

APPENDIX

ТҮРЕ	CONCEPT	BLOCK	EXPLANATION
Logic	Conditional instructions	tif do	Perform a specific task based on the evaluation of a condition.
Logic	Comparison operators		 To compare certain values: The equality test is performed with the = operator while the difference test is performed with the ≠ operator Strict comparisons are made with the < and > operators. Broad comparisons are made ith the <= and => operators.
Logic	Boolean types	true	The boolean type defines two states: true and false. The associated keyword are therefore «true» and «false».
Logic	Boolean logic operators	not	 And : return «true» if both values are «true» Or : return «true» if at least one of the two values is «true» Not : return the inverse of the boolean.
Logic	Value	null	A variable is «null» if it is null or contains no value.
Loops	Iteration	repeat 10 times do	Repeat a sequence of instructions x times, where x is the number of iterations.
Loops	Iteration	do	Create a loop that executes an instruction as long as a condition is not checked.
Loops	Iteration	count with in from 01 to 010 by 01 do	Repeat a specific sequence a number of times. Example : count i from 1 to 10 by 1 (this instruction allows to count the variable i from 1 to 10 by incrementing by 1).

ΤΥΡΕ	CONCEPT	BLOCK	EXPLANATION
Loops	Iteration	for each item 🚺 in list 🕨 do	Repeat certain instructions for each item in a list.
Loops	Iteration	break out 🔹 of loop	Exit the current loop (for/repeat/count) and go directly to the next instruction.
Math	Parameter setting		Specify a number for a function.
Math	Operator		 +: Add two numbers -: Substract the second number from the first number *: Multiply the two numbers /: Divide the first number by the second number.
Math	Operator	0 is even v	Checks if a number is • Even: exactly divisible by 2 • Odd: not exactly divisible by 2 • Prime: a prime number (2, 3, 5) • Whole: an integer • Positive: >= 0 • Negative: <= 0 • Divisible by: un nombre divisible par
Math	Operator	round v (3.1)	Return the lower integer (round down), upper integer (round up) or the nearest integer (round).
Math	Operator	sum of list	Return : • Sum: the sum of a list • Min: the minimum of a list • Max: the maximum of a list • Average: the average of a list • Median: the central value of a list • Mode: the most frequent value in a list • Standard deviation: the standard deviation of a list • Random item: a random number from the list.
Math	Operator	remainder of 64 ÷ 10	Calculate the remainder in an integer division.
Math	Operator	constrain 0 50 low 0 1 high 0 100	Return a random integer in the interval.

TYPE	CONCEPT	BLOCK	EXPLANATION
Niryo One	Movement	Move Joints j1 0 j2 0 j3 0 j4 0 j5 0 j6 0	Move the robot axes at the specified angles.
Niryo One	Movement	Move Pose x () y () z () roll () pitch () yaw ()	Move the robot to the specified position and orientation.
Niryo One	Movement	Shift pos. x v by 0	Offset a coordinate (orientation or position) to a value.
Niryo One	Calibration	Calibrate motors (auto)	Calibrate the robot automatically.
Niryo One	Calibration	Calibrate motors (manual)	Calibrate the robot manually.
Niryo One	Parameter setting	Set Arm max. speed to 1100 %	Set the maximum speed of the robot.
Niryo One	Programming	Wait for 10 seconds	Pause execution for the specified number of seconds.
Niryo One	Programming	Activate v learning mode	Activate or deactivate the learning mode.
Niryo One	Tools	Change tool to (gripper 1 •	Change the tool type.
Niryo One	Tools	gripper 1 🔹	A variable that contains the grippers available for Niryo One.
Niryo One	Tools	Detach current tool	Detach the gripper.
Niryo One	Tools	Open Gripper [gripper 1 v] at speed 2/5 v	Open the gripper at a specific speed.
Niryo One	Tools	Close Gripper (gripper 1 v at speed 2/5 v	Close the gripper at a specific speed.
Niryo One	Tools	Pull air with Vacuum Pump (Pull air with the vacuum pump to catch an object.
Niryo One	Tools	Push air with Vacuum Pump (vacuum pump 1 •	Push air from the vacuum pump to release the object.
Niryo One	Tools	Setup Electromagnet (electromagnet 1 •) with pin (1 •	Configure the electromagnet by specifying the connection pin.
Niryo One	Tools	Activate Electromagnet (electromagnet 1 •) with pin (1A •)	Activate the electromagnet by specifying the connection pin.
Niryo One	Tools	Deactivate Electromagnet (electromagnet 1 ·) with pin (1A ·)	Deactivate the electromagnet by specifying the connection pin.

ΤΥΡΕ	CONCEPT	BLOCK	EXPLANATION
Niryo One	Inputs/Outputs		A variable that contains the pins available on the Niryo One.
Niryo One	Inputs/Outputs	Set Pin (1A T) to state HIGH T	Set the digital pin to HIGH or LOW.
Niryo One	Inputs/Outputs	Set Pin (1A) to mode INPUT V	Set the pin as an: • INPUT • OUTPUT.
Niryo One	Inputs/Outputs	state (HIGH T)	Set the digital pin to HIGH or LOW.
Niryo One	Inputs/Outputs	Get Pin (1A v state	Return the value of the pin.
Niryo One	Inputs/Outputs	Set 12V Switch (SW1) to state (HIGH)	Set the switch (SW1/SW2) to 1 (HIGH) or 0 (LOW).
Niryo One	Programming	Comment :	Add a comment.
Niryo One	Programming	Break Point	Add a break point. This will pause the program. To resume execution, press the «play» button.
Variables	Variables	Create variable	Create and name a new variable. When you create a variable, the blocks corresponding to the variable are displayed.
Variables	Variables	set variable v to k	Assign a value to a variable.
Variables	Variables	change variable by 1	Change the value of a variable.
Variables	Variables	variable	Return the variable.
Lists	Lists	create empty list	Create an empty list.
Lists	Lists	create list with	Create a list by adding elements.
Lists	Lists	create list with item p repeated 5 times	Create a list with an item repeated x times.
Lists	Lists	length of	Return the lenght of the list.
Lists	Lists	in list Ist in the first in occurrence of item	Return the position of the: • First occurence of the item in a list • Last occurence of the item in a list.

TYPE	CONCEPT	BLOCK	EXPLANATION
Lists	Lists	in list (list v get v # v k	 Return: # : the n item in the list # from end : the n item in the list from the end of it First : the first item in the list Last : the last item in the list Random : a random item in the list.
Functions	Functions	input name: X	Create a procedure with arguments (input parameters).
Functions	Functions	input name: X inputs	Create a procedure with arguments (input parameters) and an output (return).



LEARNING MODE: CORRECTION

- Activate the learning mode of the robot
- Move the robot with your hand to a position of your choice
- Hold the robot in this position and click on « UPDATE VALUES » before releasing the robot



You can also press the upper button on the robot to save the value

- Deactivate the « learning mode »
- Execute a movement to this position by clicking on « MOVE AXES »

Did the robot make the desired movement?

The robot made the desired movement: indeed, the learning mode is designed to save the positions and sequences manually determined by the operator and then reproduce them.

• We would like to put the robot in a « top » position as shown in the figure below:



Which axis will reach its maximum angle?

The joint 3 will reach its maximum angle of 90°.

- Activate the « learning mode » and move the robot with your hand to this position
- Hold the robot in this position, select « UPDATE VALUES » and then save it with the name « TOP »
- Select this position and execute it by clicking on « MOVE AXES »

BLOCKLY: CORRECTION

We want to move the Niryo One to the two positions P1 and P2 :

P1 = [x = -0.03; y = -0.156; z= 0.48; roll= -0.58; pitch = -0.58; yaw = -0.145]

P2 = [x = -0.136 ; y = -0.133 ; z = 0.255 ; roll= -0.081 ; pitch = 0.744 ; yaw = -2.535]

After reading the appendix, identify the block that allows you to move the robot to a position with Cartesian coodinate x y and z.

Move Pose x 0 y 0 z 0 roll 0 pitch 0 yaw 0

• Reproduce the following sequence in Niryo One Studio

Move Pose x -0.03 y -0.156 z 0.48 roll -0.58 pitch -0.58 yaw -0.145 Move Pose x -0.136 y -0.133 z 0.255 roll -0.081 pitch 0.744 yaw -2.532

- Deactivate the « learning mode » and execute the program
- Add the block to your sequence as shown in the illustration below :

repeat 10 times

- do Move Pose x -0.03 y -0.156 z 0.48 roll -0.58 pitch -0.58 yaw -0.145 Move Pose x -0.136 y -0.133 z 0.255 roll -0.081 pitch 0.744 yaw -2.532
- Deactivate the « learning mode » and execute the is new sequence

Explain the difference between the first and the second sequence and conclude on the role of the added block.

In the first code, the robot moved to a first position then to a second one. In the second code, the robot followed the same path and repeated this sequence 10 times.

We want to automate the task of activating and deactivating the « learning mode ».

Which block should be used to perform this operation? Add this block at the beginning and at the end of your program, then execute it.

Activate V learning mode

As explained in the appendix, the function of this block is to activate or deactivate the learning mode.

A sequence of instructions represents the program to be executed by the robot.

Keeping the program developed in the previous sections, click on the « import/export » button.



Name	
Description	
Blockly XML	
<xml xmlns="http://www.w3.org/1999/xhtml"><block id="2H-t.aSy+N:^i`w+*a?r" type="niryo_one_activate_learning_mode" x="-770" y="-710"><field name="LEARNING_MODE_VALUE">0<next><block id="Q9PmGur00P6[Y]=1%WSf" type="controls_repeat_ext"><value name="TIMES"><shadow< td=""><td>÷ //</td></shadow<></value></block></next></field </block></xml>	÷ //
SAVE	

A sequence is characterized by:

- Its name (by default, it will be « sequence » + ID)
- Its description

We want to execute the following program:



Knowing that:

- The positions of the robot are in joint coordinates
- The initial position of the robot is [0;0;0;0;0;0]
- P1 = [-1.215;-0.333;-0.867;-3.053;-0.014;0.127]
- P2 = [-1.377; 0.22; -0.281; -3.053; -0.031; 1.549]

Translate the previous diagram into text.

First, the learning mode is deactivated. Then, the speed of the robot is set at 100% and the robot goes to its initial position.

The robot goes to P1, waits 2 seconds, then goes to P2, and repeats 3 times this operation. Once the 3 repetitions achieved, the robot will go to its initial position and the learning mode will be activated.



Move Joints j1 0 j2 0 j3 0 j4 0 j5 0 j6 0 Set Arm max. speed to 100 % set Counter to 0 repeat while Counter 🔹 < • 3 do Move Joints j1 1.215 j2 0.333 j3 0.867 j4 3.053 j5 0.014 j6 0.127 Wait for 2 seconds Move Joints j1 1.337 j2 0.22 j3 0.281 j4 3.053 j5 0.031 j6 1.549 Wait for 2 seconds change Counter v by 1 Move Joints j1 0 j2 0 j3 0 j4 0 j5 0 j6 0 Activate v learning mode

Taking into account the counter variable and the loop block, propose a block to replace these two blocks.



• Replace this variable and loop blocks with this new block

• Execute the sequence and **comment**.

The variable Counter acts as a counter for iteration. Loop blocks such as «repeat» or «count with» can replace this variable and the «repeat while».