Tic tac toe demonstrator

User instructions

Table of contents

Turning on the robot....3
Basic explanation of the program....3
Robot....3
Arduino....5
initialise()....6
playersTurn()....7
printGame()....7
checkGame()....8
endGame()....8
robotsTurn()....9
checkPossibilities()....10
checkBlockers()....12
randomPlay()....14
playPos(x,y)....15
initializeRobotBits()....15
Game play....16
Shutdown of the robot....16

Turning on the robot

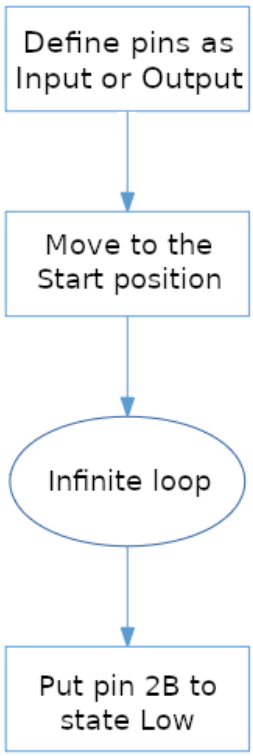
Connect the robot with the power supply given. Toggle the robot switch to power it on. When the led at the back of the robot become blue or green, the robot is ready. You can then calibrate the robot.

Then open the Niryo studio application, connect to the robot, and start the program Tic-tac-toe.

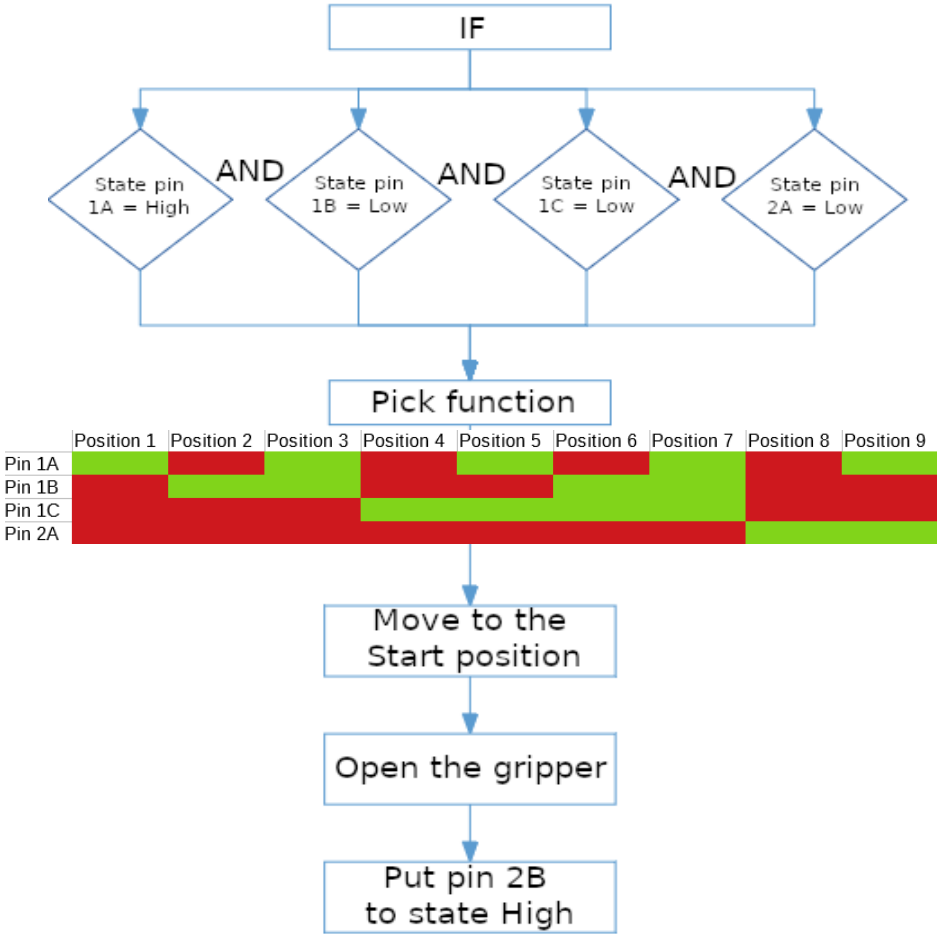
Finally, launch the program « tic-tac-toe_game » on the Arduino board.

Basic explanation of the program

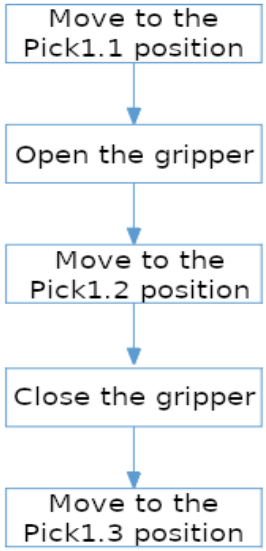
Robot



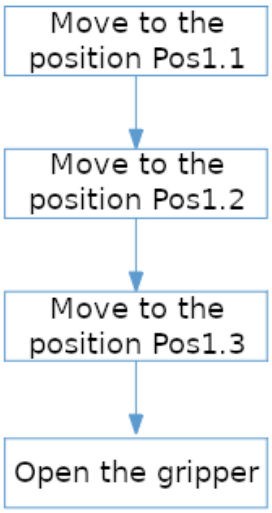
The robot can play on 9 different positions, we use a boolean logic to define which position to play. The 9 positions are defined according to the following table:



Each position are then defined according to the same scheme, and call the corresponding function (Pos1, Pos2, Pos3, …) according to the pins state.

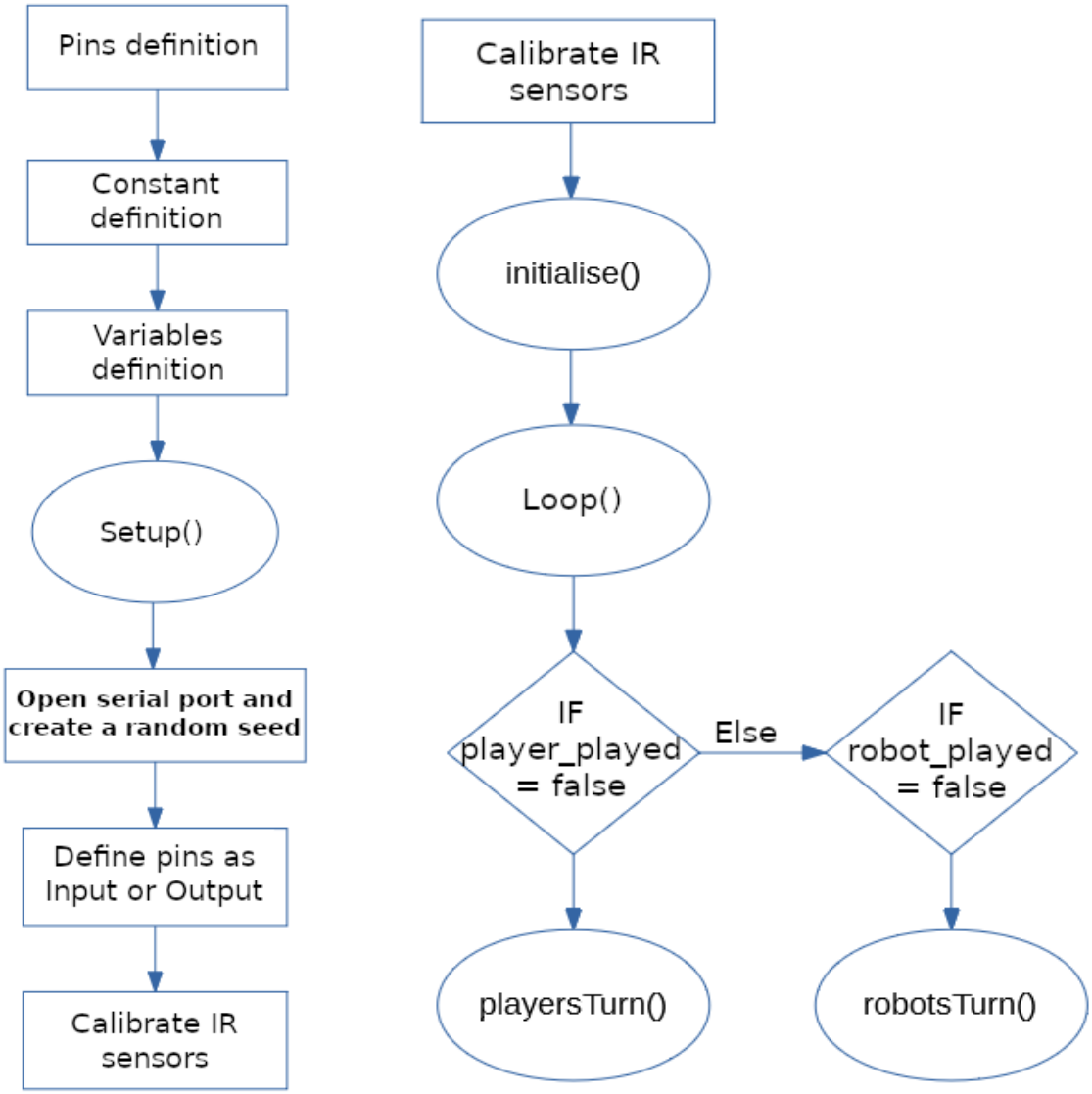


Description of the Pick function

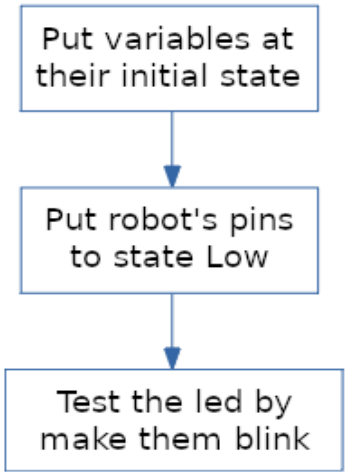


Description of the Pos1 function

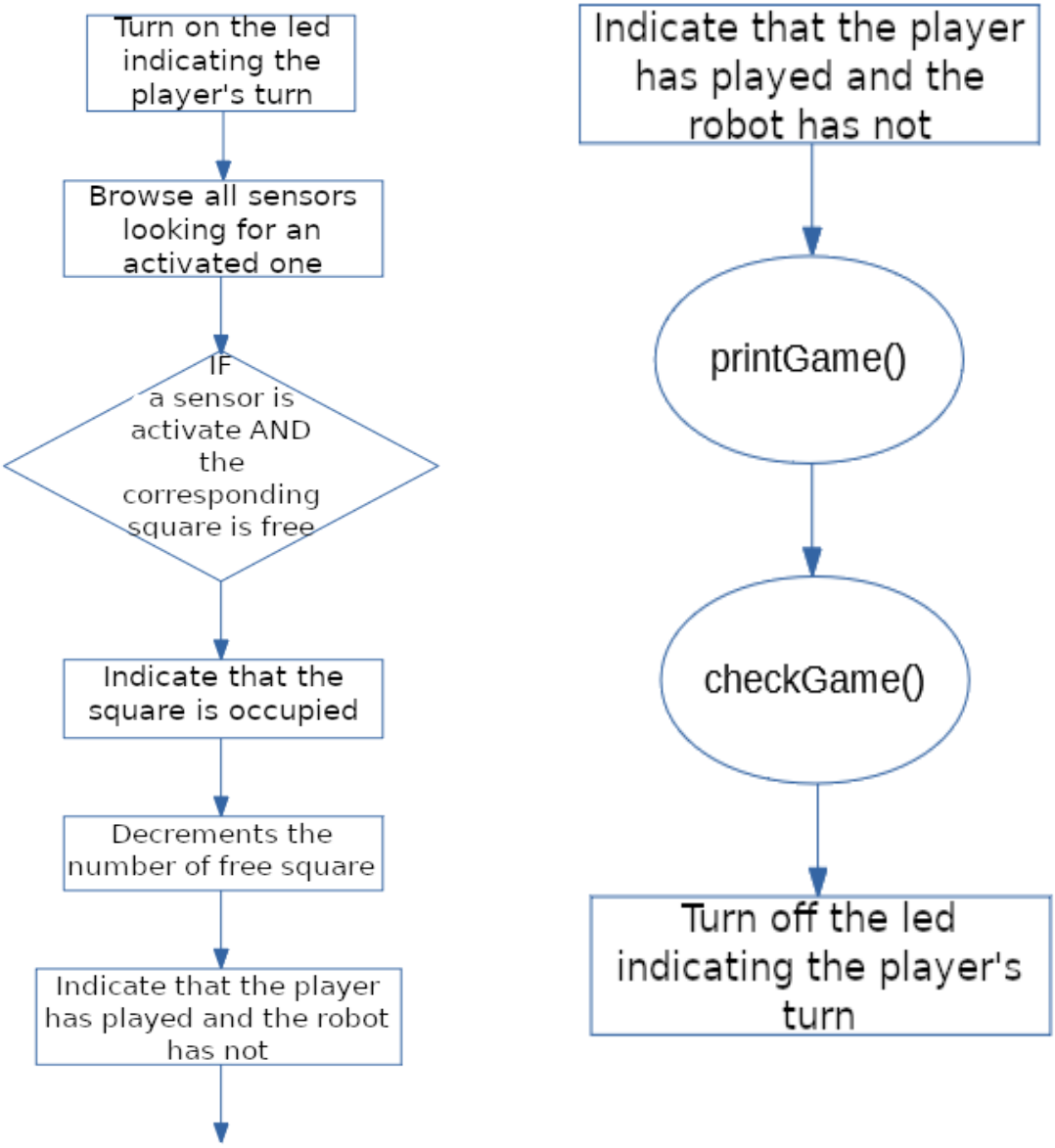
Arduino


Pins definition
Constant definition
Variables definition
Setup()
Open serial port and create a random seed
Define pins as Input or Output
Calibrate IR sensors
Calibrate IR sensors
initialise()
Loop()
IF player_played = false
Else
IF robot_played = false
playersTurn()
robotsTurn()

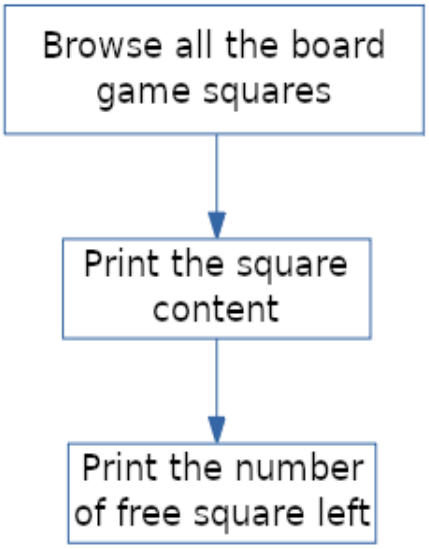
initialise()


Put variables at their initial state
Put robot's pins to state Low
Test the led by make them blink

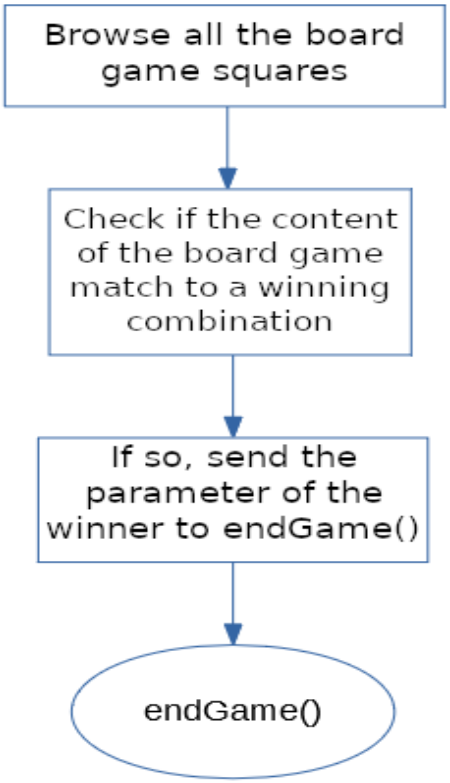
playersTurn()



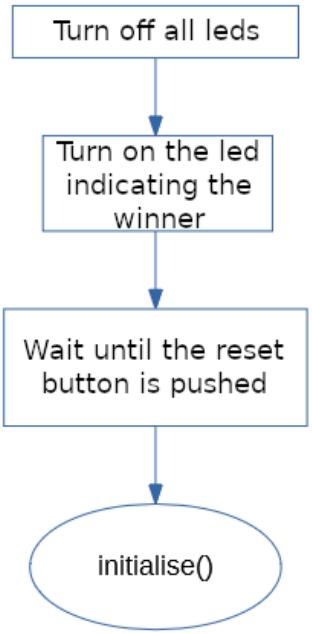
printGame()



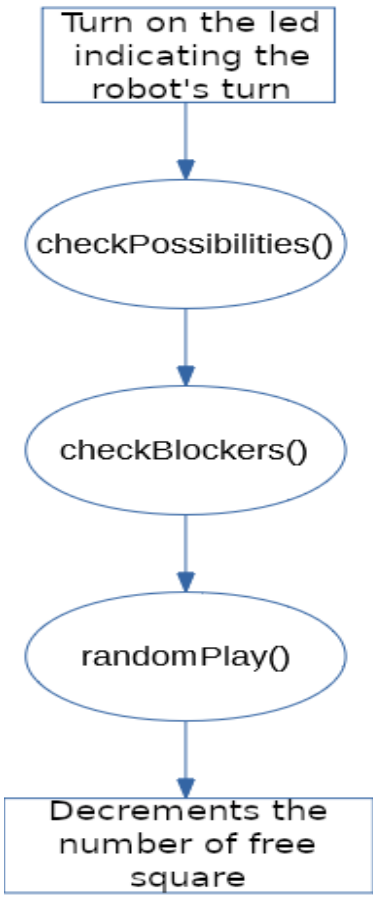
checkGame()

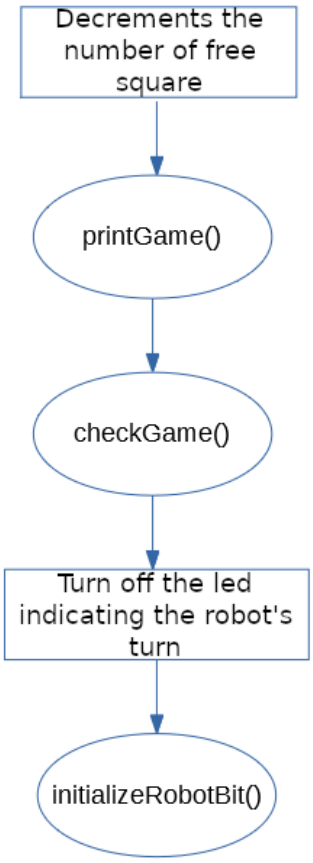


endGame()

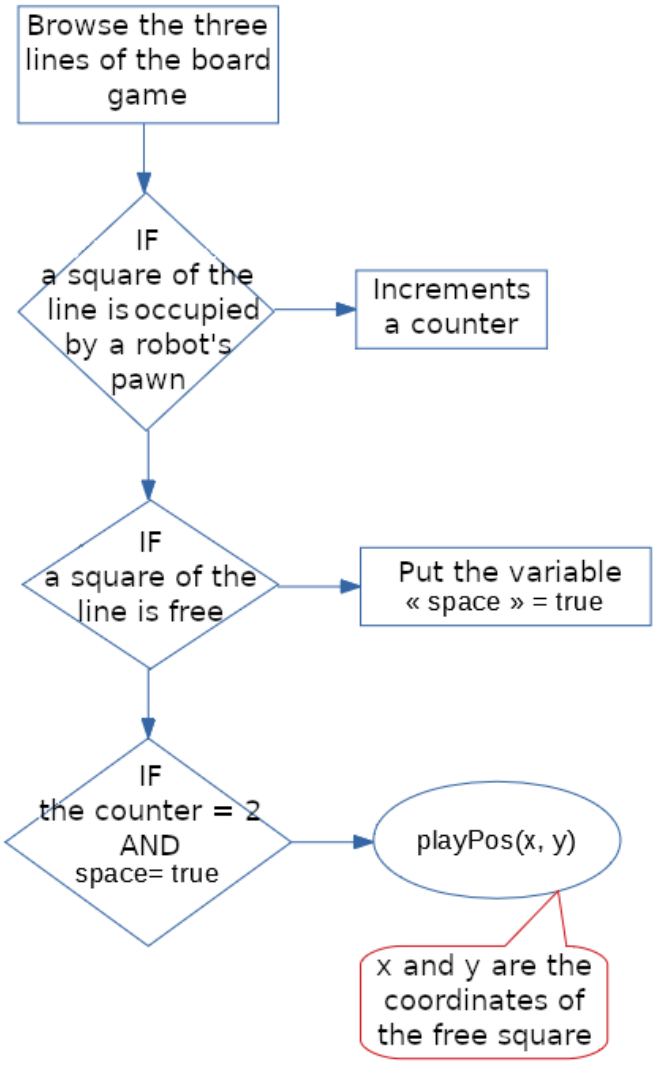


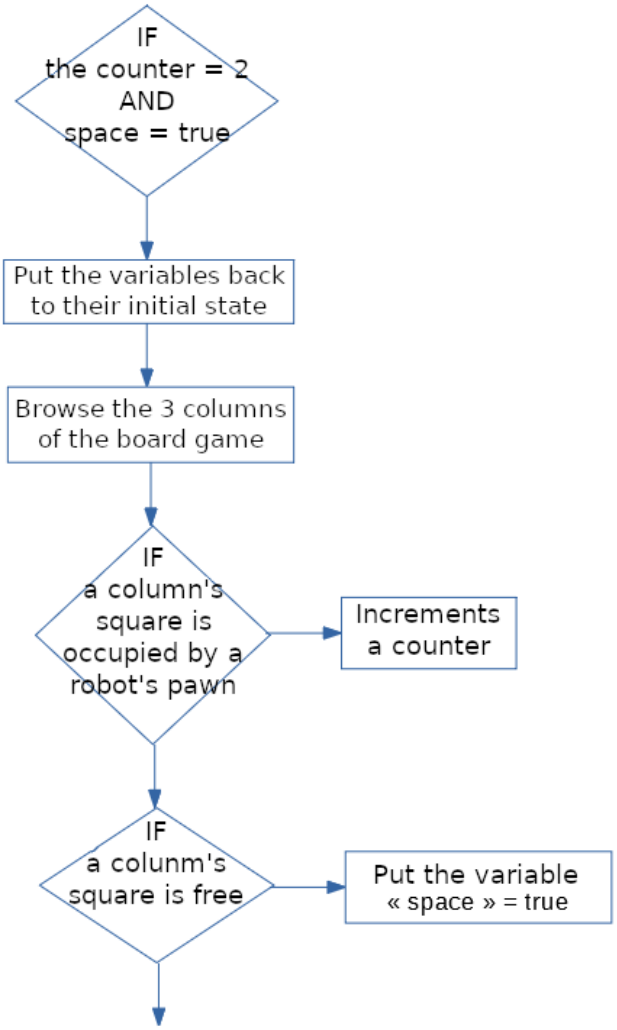
robotsTurn()

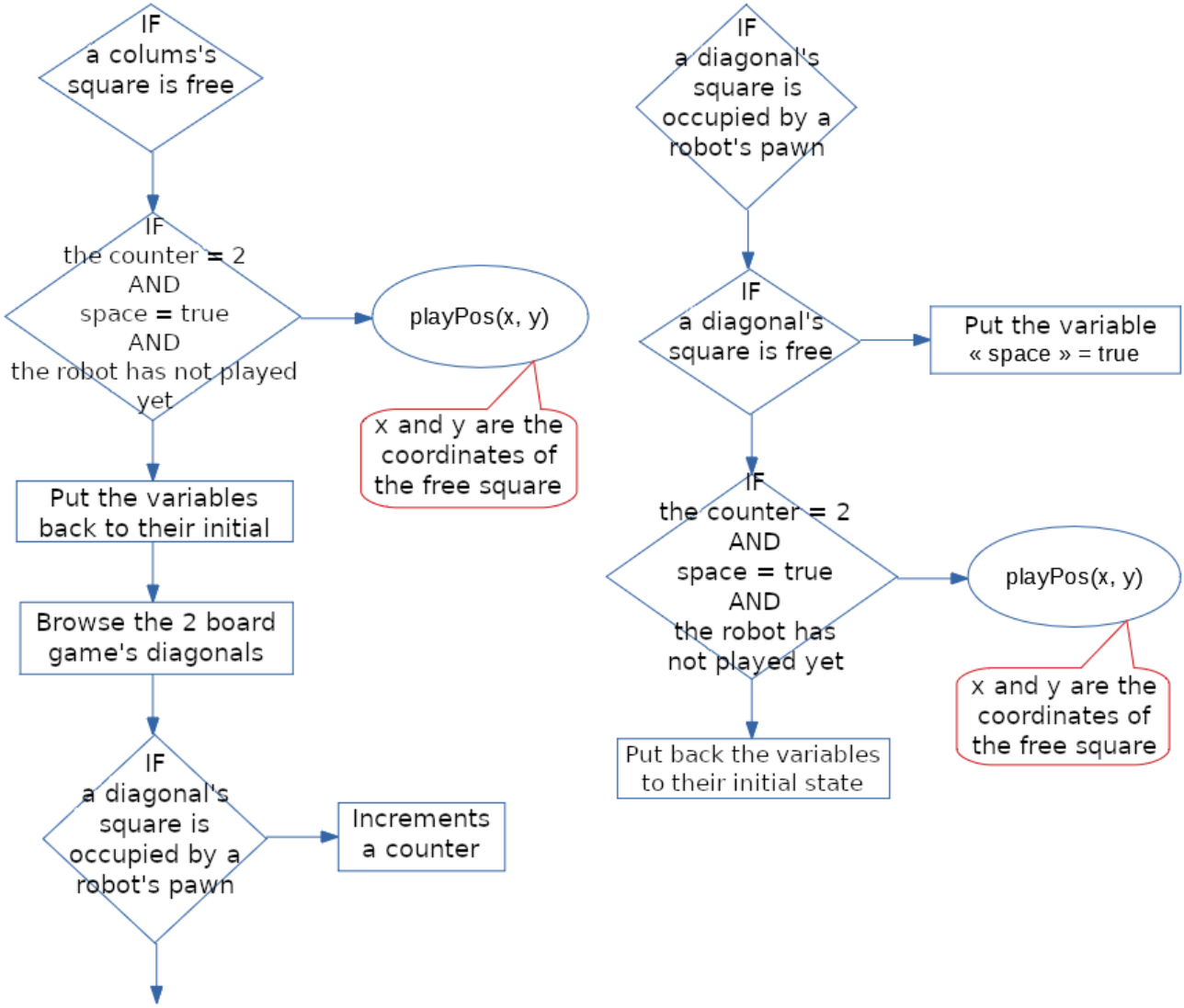

Turn on the led indicating the robot's turn
checkPossibilities()
checkBlockers()
randomPlay()
Decrements the number of free square


Decrements the number of free square
printGame()
checkGame()
Turn off the led indicating the robot's turn
initializeRobotBit()

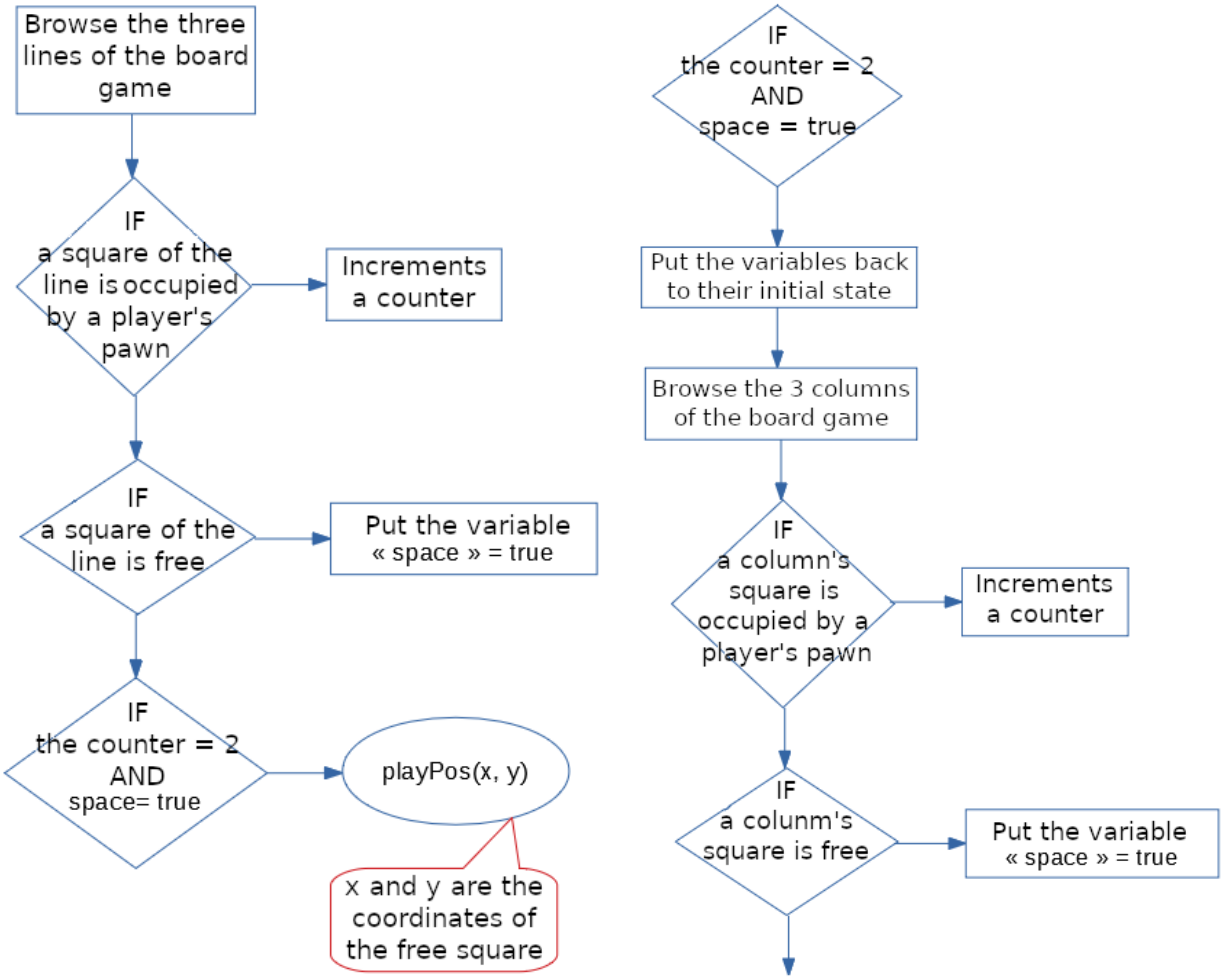
checkPossibilities()

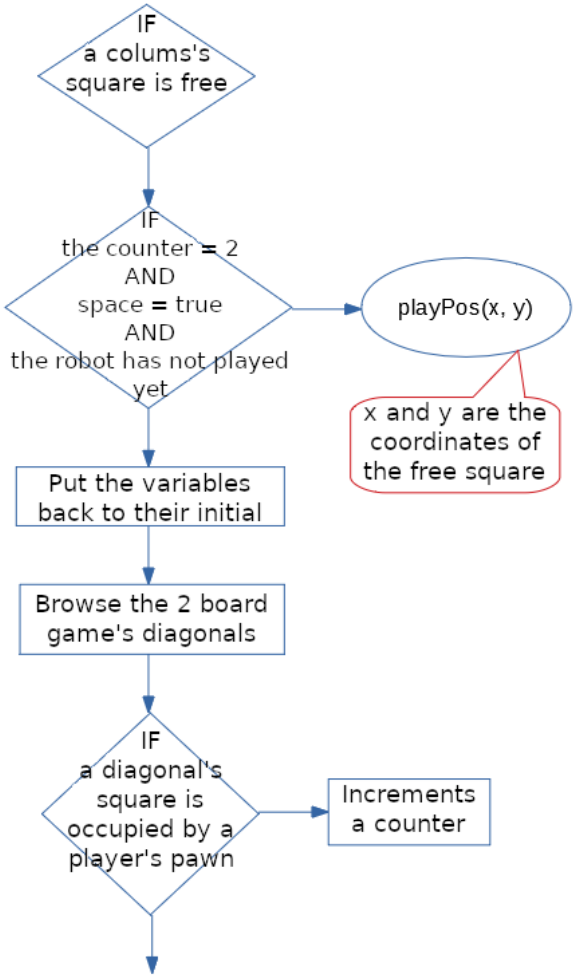


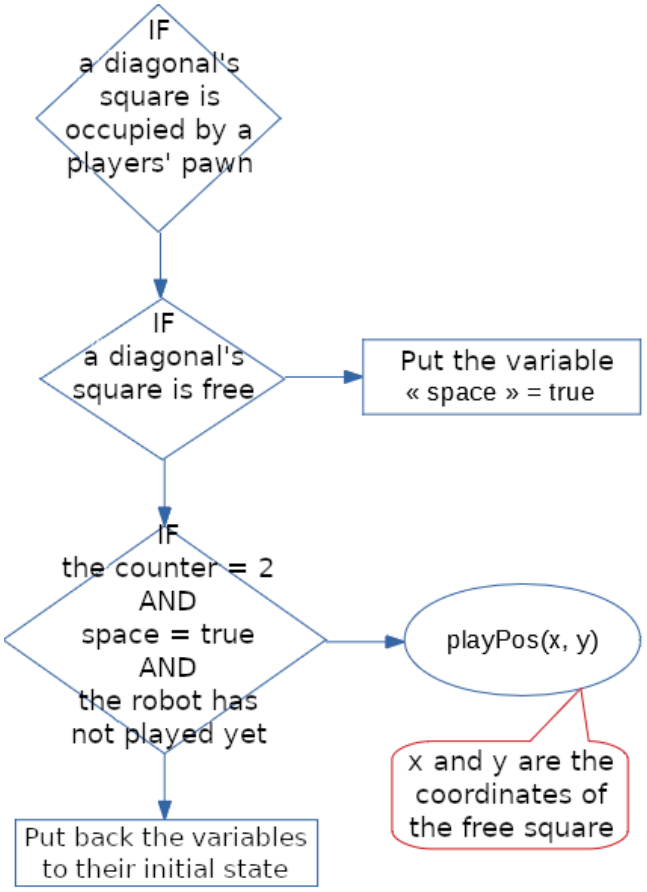



IF
a colums's
square is free
IF
the counter = 2
AND
space = true
AND
the robot has not played
yet
playPos(x, y)
x and y are the
coordinates of
the free square
Put the variables
back to their initial
Browse the 2 board
game's diagonals
IF
a diagonal's
square is
occupied by a
robot's pawn
Increments
a counter
IF
a diagonal's
square is
occupied by a
robot's pawn
IF
a diagonal's
square is free
Put the variable
« space » = true
IF
the counter = 2
AND
space = true
AND
the robot has
not played yet
playPos(x, y)
x and y are the
coordinates of
the free square
Put back the variables
to their initial state

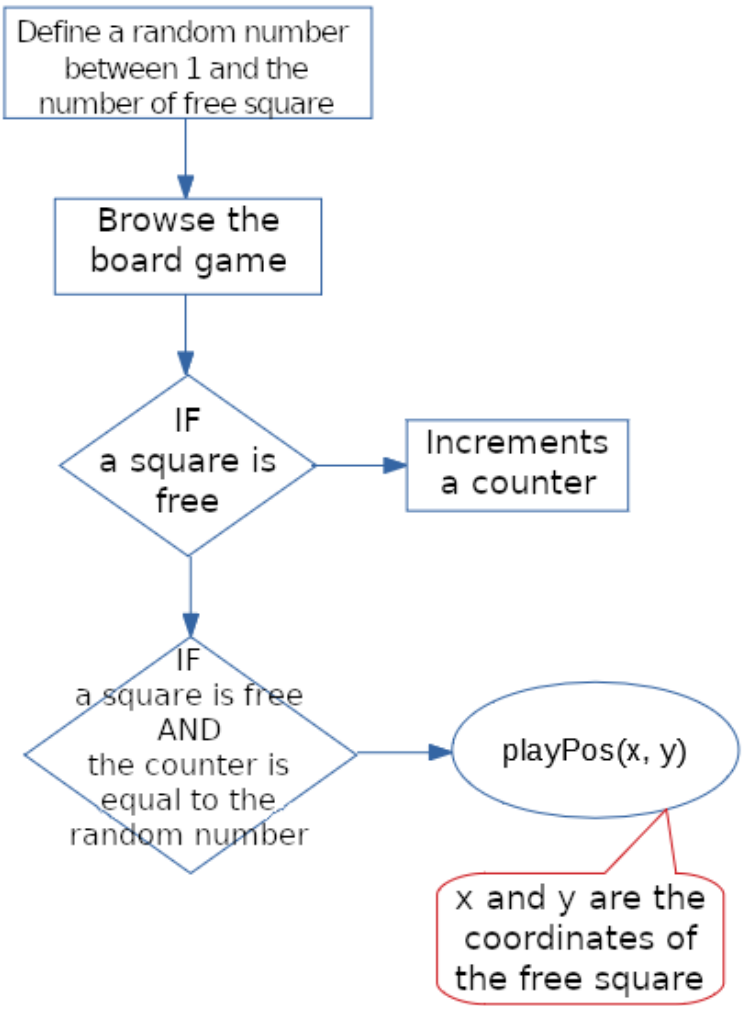
checkBlockers()



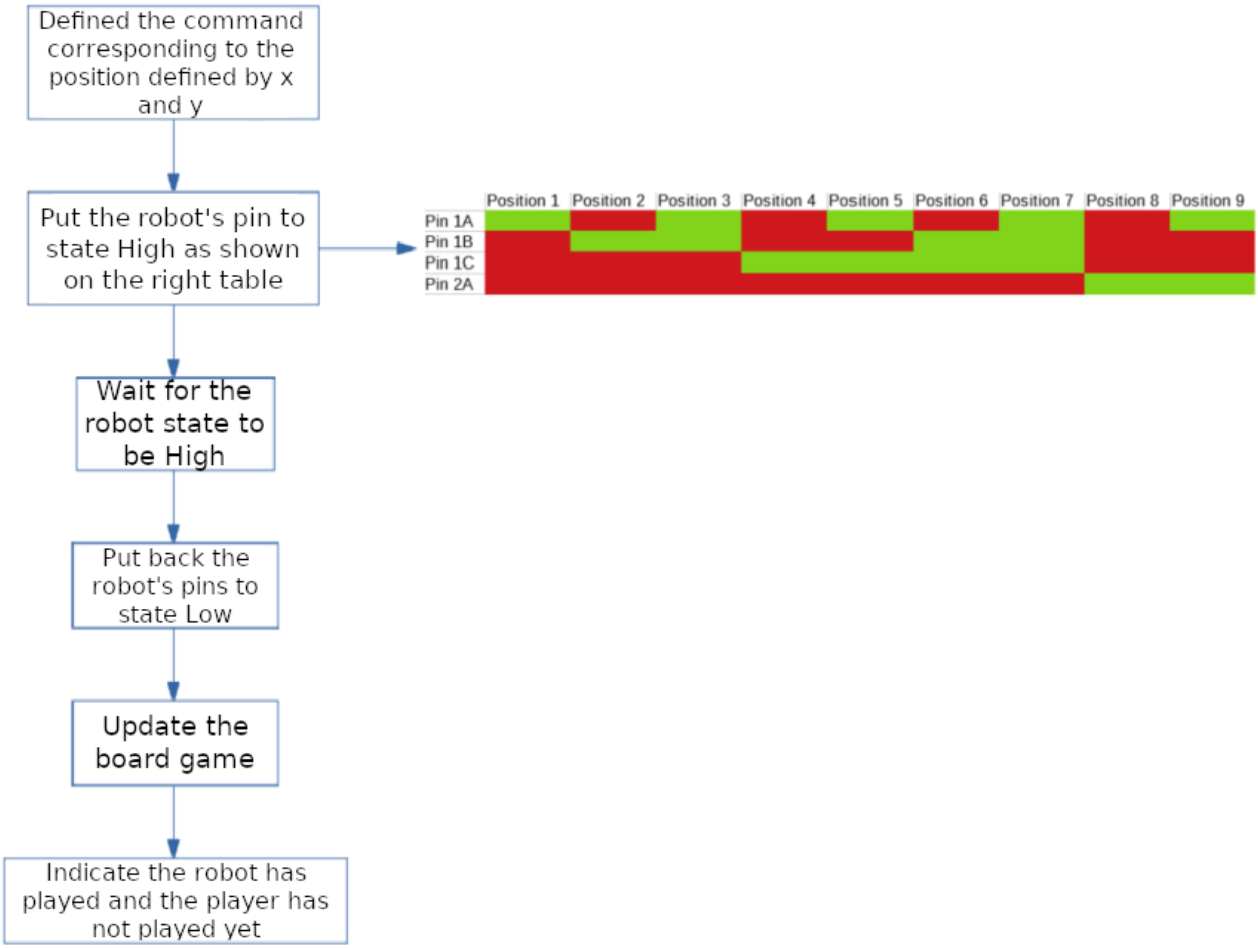

IF
a colums's
square is free
IF
the counter = 2
AND
space = true
AND
the robot has not played
yet
playPos(x, y)
x and y are the
coordinates of
the free square
Put the variables
back to their initial
Browse the 2 board
game's diagonals
IF
a diagonal's
square is
occupied by a
player's pawn
Increments
a counter


IF
a diagonal's
square is
occupied by a
players' pawn
IF
a diagonal's
square is free
Put the variable
« space » = true
IF
the counter = 2
AND
space = true
AND
the robot has
not played yet
playPos(x, y)
x and y are the
coordinates of
the free square
Put back the variables
to their initial state

randomPlay()


Define a random number between 1 and the number of free square
Browse the board game
IF a square is free
Increments a counter
IF a square is free AND the counter is equal to the random number
playPos(x, y)
x and y are the coordinates of the free square

playPos(x,y)



initializeRobotBits()

Put all robot’s pins to state Low.

Game play

When you start the program on the robot and on the Arduino board, the sensors will calibrate, then all the leds will blink once, and it is first the player's turn. The green led (the one indicating the player's turn) will turn on and the program will wait for the player to place a pawn. When a pawn is placed on a square, the corresponding sensor will be activated. We then print the board game in the terminal, and we browse all the squares to check for a winner. If there is no winner, the green led will turn off and it is the robot's turn.

The red led (the one indicating the robot's turn) will turn on, the program will browse the board game and check if there is a victory possibility. If there is one, the robot will play on this square. If not, the program will check if the player has a possibility to win, and if it is possible to block it. If it is possible, the robot will play on this position, if not, the robot will play on a random square.

The program will then print the board game on the terminal, and we browse all the squares to check for a winner. If there is no winner, the red led will turn off, and the robot's pins will be put back to low state, and it is now the player's turn.

If there is a winner the corresponding led will turn on.

Finally, the programme will wait for the reset button to be pushed, when it is, the program will start again and it will be player's turn.

Shutdown of the robot

To shutdown the demonstrator, you just have to stop the program on the robot on Niryo Studio, you then need to press the top button and wait for the led to turn purple (after about 3 seconds), then wait for the led to turn red and when it is, you can switch off the robot and unplug it.