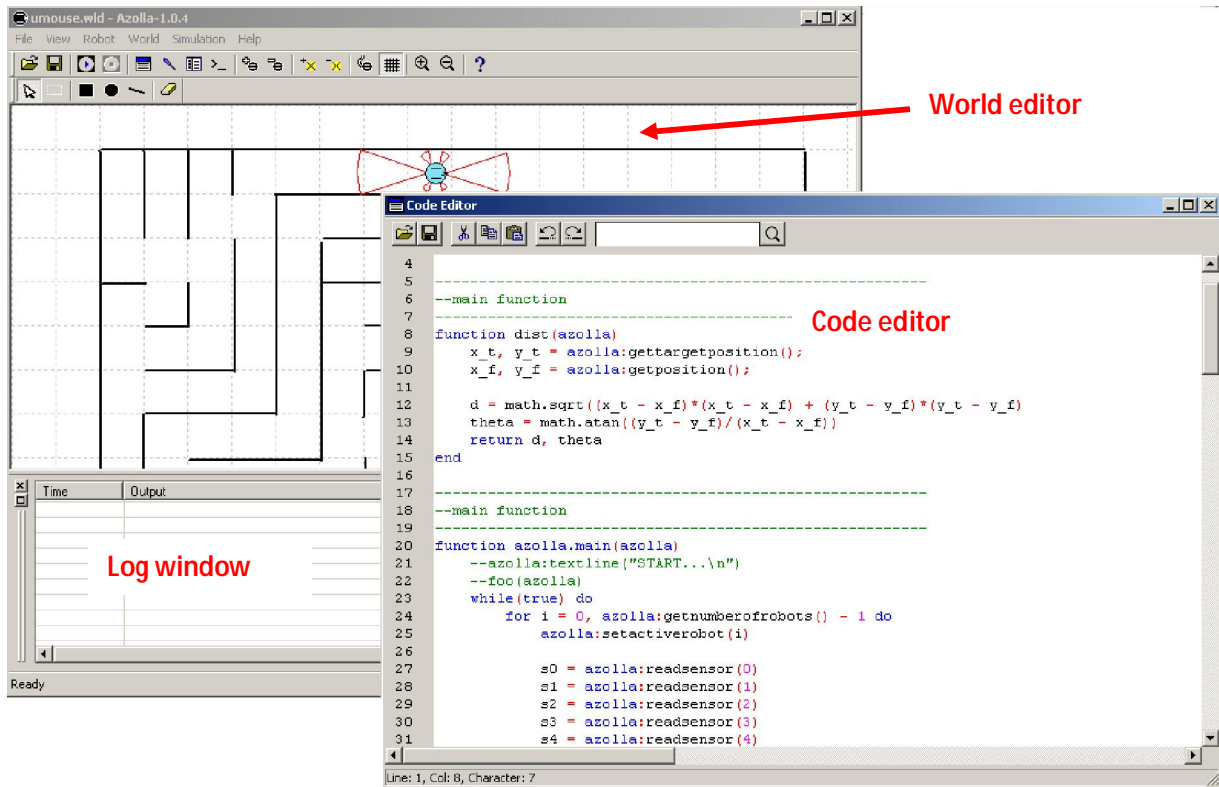
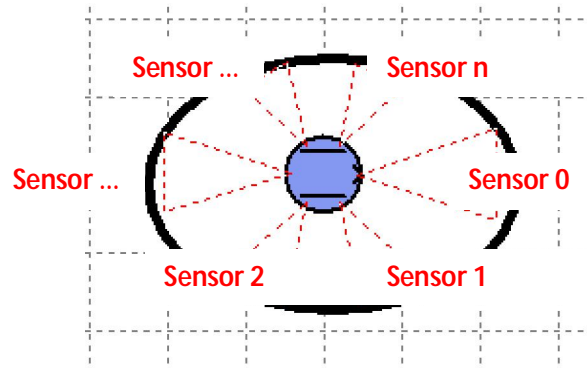


AZOLLA 1.0.4 GETTING STARTED



Reading sensors



Sensor indexing is started from front side. Sensor index number is increasing in clockwise direction. The following instruction will read sensor value (in cm) and the results will be saved in a variable:

```
front = azolla:readsensor(0)
```

Moving the robot

The robot has two wheels in left and right side. The robot will move using differential steering method. The following instruction will set the speed of both wheels:

```
azolla:setspeed(left_speed, right_speed)
```

Variable `left_speed` and `right_speed` are float numbers. The speed is in cm/s.

Simple Algorithm for Wall Following

A very simple algorithm to follow walls in its left and right is by making sure that distance between left and right wall is the same.

- If the robot is too close to the right wall, then turn left
- If the robot is too close to the left wall, then turn right

In the code listed below, the robot should keep its distance from left and right wall. Maximum difference between distance of left wall and right wall is 2 cm.

```
function azolla.main(azolla)
  azolla:setspeed(20,20)
  while true do
    a = azolla:readsensor(1)
    b = azolla:readsensor(5)
    if (a - b > 2) then
      azolla:setspeed(7,5)
    end

    if (a - b < -2) then
      azolla:setspeed(5,7)
    end

    if( (a - b > -2) and (a - b < 2) ) then
      azolla:setspeed(5,5)
    end
    azolla:stepforward()
  end
end
```

Programming Structure

Every program must have main function. In main function, there should be main loop. Every time the loop ends, `azolla:stepforward()` should be executed.

```
function azolla.main(azolla)
    while true do
        --
        -- write your code here
        --
        azolla:stepforward()
    end
end
```

Multiple Robot Programming

For multiple robot programming, before giving command, such as reading sensor or setting speed, you should activate the robot first using command: `azolla:setactiverobot(index)` Index is started from zero.

The code listed below is to make several robots to follow the wall.

```
function azolla.main(azolla)
    azolla:setspeed(20,20)
    while true do
        for i = 0, azolla:getnumofrobots() - 1 do
            azolla:setactiverobot(i)
            a = azolla:readsensor(1)
            b = azolla:readsensor(5)
            if (a - b > 2) then
                azolla:setspeed(7,5)
            end

            if (a - b < -2) then
                azolla:setspeed(5,7)
            end

            if( (a - b > -2) and (a - b < 2) ) then
                azolla:setspeed(5,5)
            end
        end
        azolla:stepforward()
    end
end
```

When programming multiple robots, you should avoid using loop inside a loop. The best solution is to use state machine model.

Procedural Programming

Lua supports procedural programming. You can create several functions where in each function you can assign a certain task.

function `azolla.main(azolla)` will always be first called. It's the main function. Since we must always pass `azolla` as parameters in every function, the function will always be written with something like this: `function function_name(azolla, parameter1, parameter2...)`.

```
function function1(azolla)
    -- do something
    azolla:textline("Calling function 1\n")
end
```

```
function function2(azolla)
    -- do something
    azolla:textline("Calling function 2\n")
end
```

```
function function3(azolla, x, y)
    -- Accept and return several parameters
    x = x / 2
    y = y / 2
    return x,y
end
```

```
-----
--main function
```

```
function azolla.main(azolla)
    azolla:textline("START...\n")
    while(true) do
        function2()
        function2()
        x, y = function3(3, 4)
    end
end
```

Wandering Maze Using Simple P (Proportional) Algorithm

In the code listed below, the robot will travel around the maze using simple P(proportional) algorithm. The robot will read input from left and right sensor. Control signal (delta) will be calculated based on the difference between left and right sensor value. This control signal will be used to correct the speed of both wheels.

```
function azolla.main(azolla)
  azolla:textline("START...\n")
  while(true) do
    for i = 0, azolla:getnumofrobots() - 1 do
      azolla:setactiverobot(i)
      front = azolla:readsensor(0)
      left = azolla:readsensor(5)
      right = azolla:readsensor(1)

      if (front < 10) then
        azolla:setspeed(4,-4)

      else
        delta = 0.5 * (right - left)
        azolla:setspeed(4 + delta,4 - delta)
      end
    end
    azolla:stepforward()
  end
end
```

To see how it works, there are two files included for demonstration: *trinity.wrl* and *trinity-multirobots.lua*. Load the two files and run simulation.