

EE1130

Freshman Design

(rev Oct14)

Class controls II:
RobotC (Lego Mindstorm).


RobotC

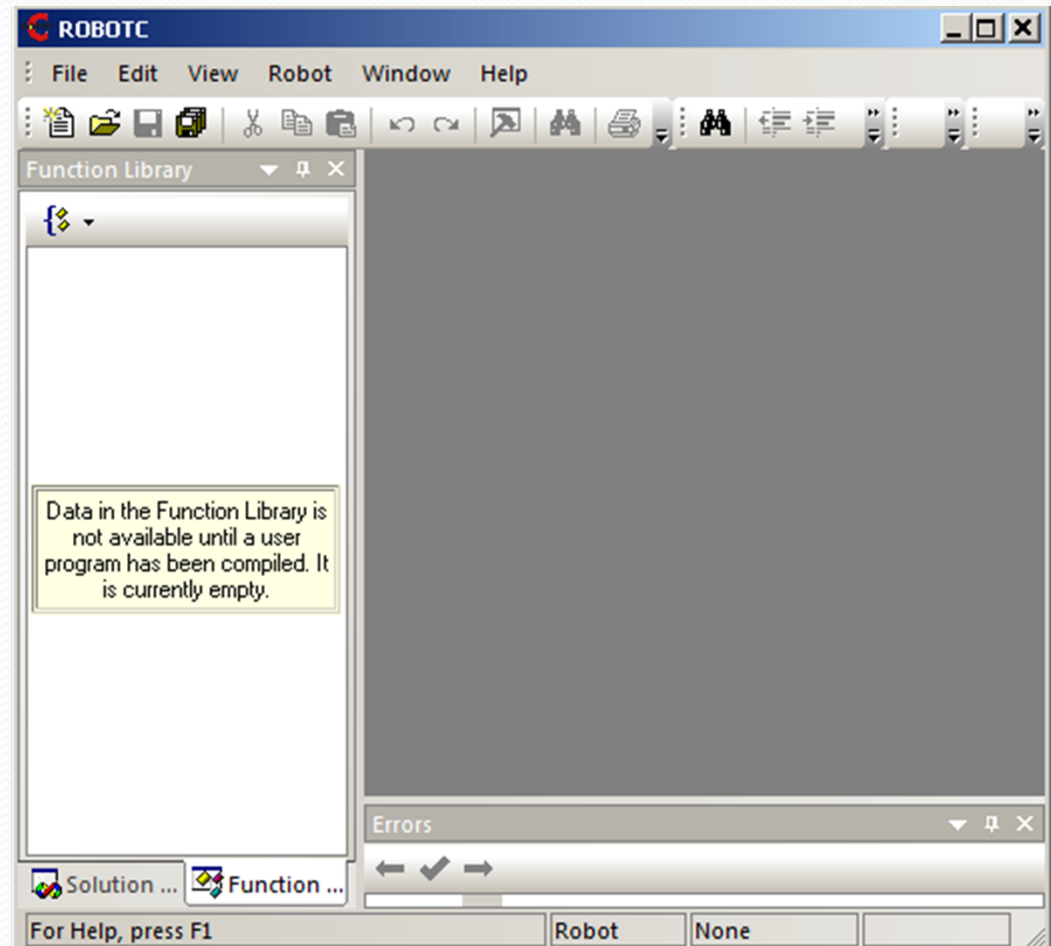
- Lego Mindstorms RobotC es una plataforma de diseño de programas para controlar un robot
 - Tiene una parte virtual donde uno implementa el programa y lo prueba con un robot virtual.
 - Tiene una parte práctica donde el programa se baja (via USB) a un robot de verdad y el programa se instala en el DSP (Digital Signal Processing unit) del Robot.



RobotC

- Instalar y abrir el Lego .

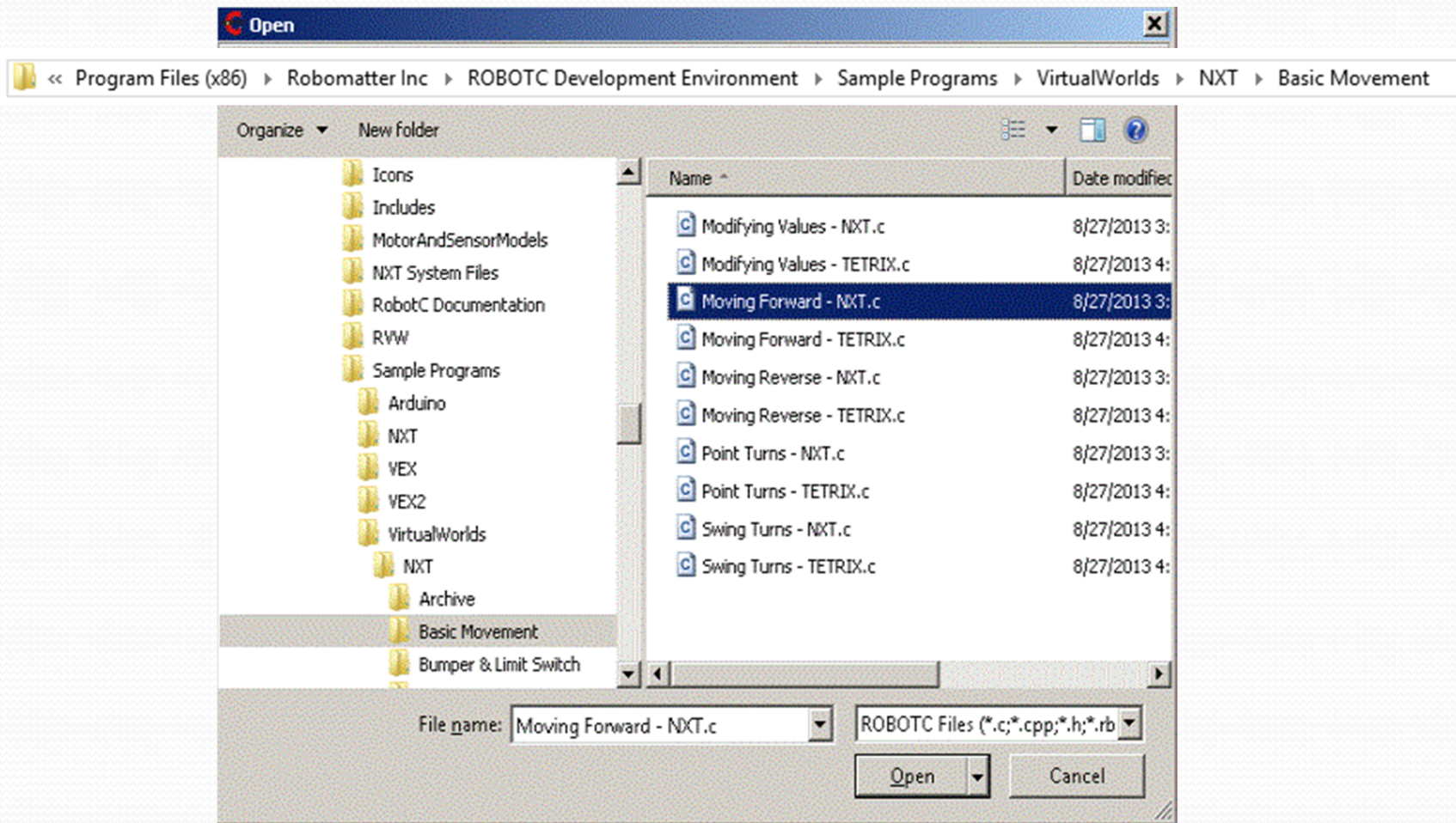
 ROBOTC Virtual Worlds - MINDSTORM



- Por defecto debe trabajar tal y como está la configuración.

RobotC

- Abran el siguiente fichero de Samples Programs
- C:\Program Files\Robomatter Inc\ROBOTC Development Environment\Sample Programs\VirtualWorlds\NXT\Basic Movement



RobotC

- Veran un codigo similar a este:

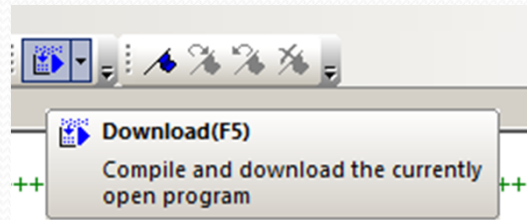
```
LEGO Start Page Moving Forward - NXT.c*
1  #pragma config(StandardModel, "RVW REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard          !!**/
3
4  /*+++++| Notes |+++++
5  Moving Forward
6  This program instructs your robot to move forward at full power for three seconds.
7  There is a two second pause at the beginning of the program.
8
9  Robot Model(s): NXT REMBOT
10
11  [I/O Port]      [Name]      [Type]      [Description]
12  Motor Port B    rightMotor  NXT Motor   Right side motor
13  Motor Port C    leftMotor   NXT Motor   Left side motor
14
15  -----*/
16
17
18  //+++++| MAIN |+++++
19  task main()
20  {
21      //wait1Msec(2000);          // Robot waits for 2000 milliseconds before executing program
22
23      // Move forward at full power for 3 seconds
24      motor[rightMotor] = 100;    // Motor on motorB is run at full (100) power forward
25      motor[leftMotor]  = 100;    // Motor on motorC is run at full (100) power forward
26      wait1Msec(3000);          // Robot runs previous code for 3000 milliseconds before moving on
27  }                               // Program ends, and the robot stops
28  //+++++|
```


RobotC

- Debemos modificar el código para que el robot sea capaz de hacer lo que le pedimos.
- Vamos a compilar y abrir el Virtual Worlds.

RobotC

- Para ellos vamos a darle a F5 para compilar y ejecutar el programa!!



- A veces sale un warning. Luego sale la ventana siguiente y le dan a Log In as Guest

A screenshot of the RobotC login screen. At the top, there are two tabs: 'Log In to CS2N' (highlighted in yellow) and 'Log In Locally'. The main area has a dark background with a grid pattern. On the left, there is a blue CD-ROM icon and the text 'LOCAL Create Account'. In the center, the text reads 'ROBOT VIRTUAL WORLDS' and 'Log in with your Local account to save progress!'. Below this are input fields for 'Username' and 'Password', and a checkbox for 'Remember Information?'. At the bottom, there are three buttons: 'Log In Locally', 'OR', and 'Log In as Guest'. A warning message at the bottom center says 'Your Progress will not be saved'.

RobotC

- Aparece la siguiente ventana:
 - Elegir un Robot
 - Elegir un ambiente de simulacion

ROBOTS

- LEGO REMBot
- REMBot (w. Touch)
- Buggy Bot
- Mammal Bot

MOVEMENT

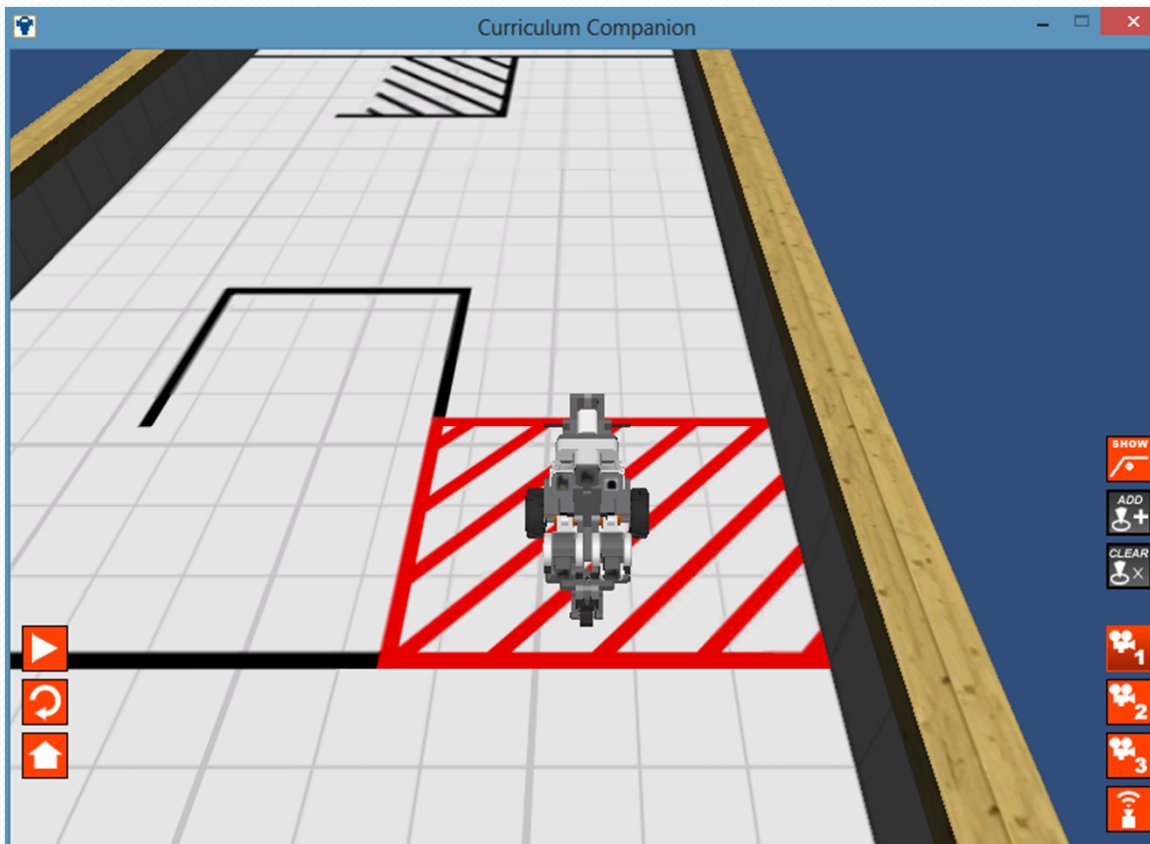
- ☆ Labyrinth Challenge

The screenshot shows the 'Curriculum Companion' window. At the top, there are navigation tabs: HOME, LOGIN, OPTIONS, BADGES, ROBOTS, MOVEMENT, SENSING, VARIABLES, REMOTE CONTROL, and UTILITY. The main content area is divided into two columns. The left column has a 'WELCOME' header and three paragraphs of instructions: 'To get started, click on the Robots tab at the top of the screen to pick a robot.', 'Then click one of the Category tabs and choose a table.', and 'Finally, click the "Start Activity" button to begin the activity.' Below the text is the Robomatter logo and the text 'Manufactured with intellectual properties from Carnegie Mellon University'. The right column features a large blue banner with the text 'ROBOT VIRTUAL WORLDS FOR NXT' and an image of five different LEGO robots. Below the banner is a large red button that says 'CLICK HERE to access training materials for ROBOT VIRTUAL WORLDS'. At the bottom of the window, there is a footer with the text 'Curriculum Companion for NXT v3.1.1 (C) 2013 Robomatter Inc.'

RobotC

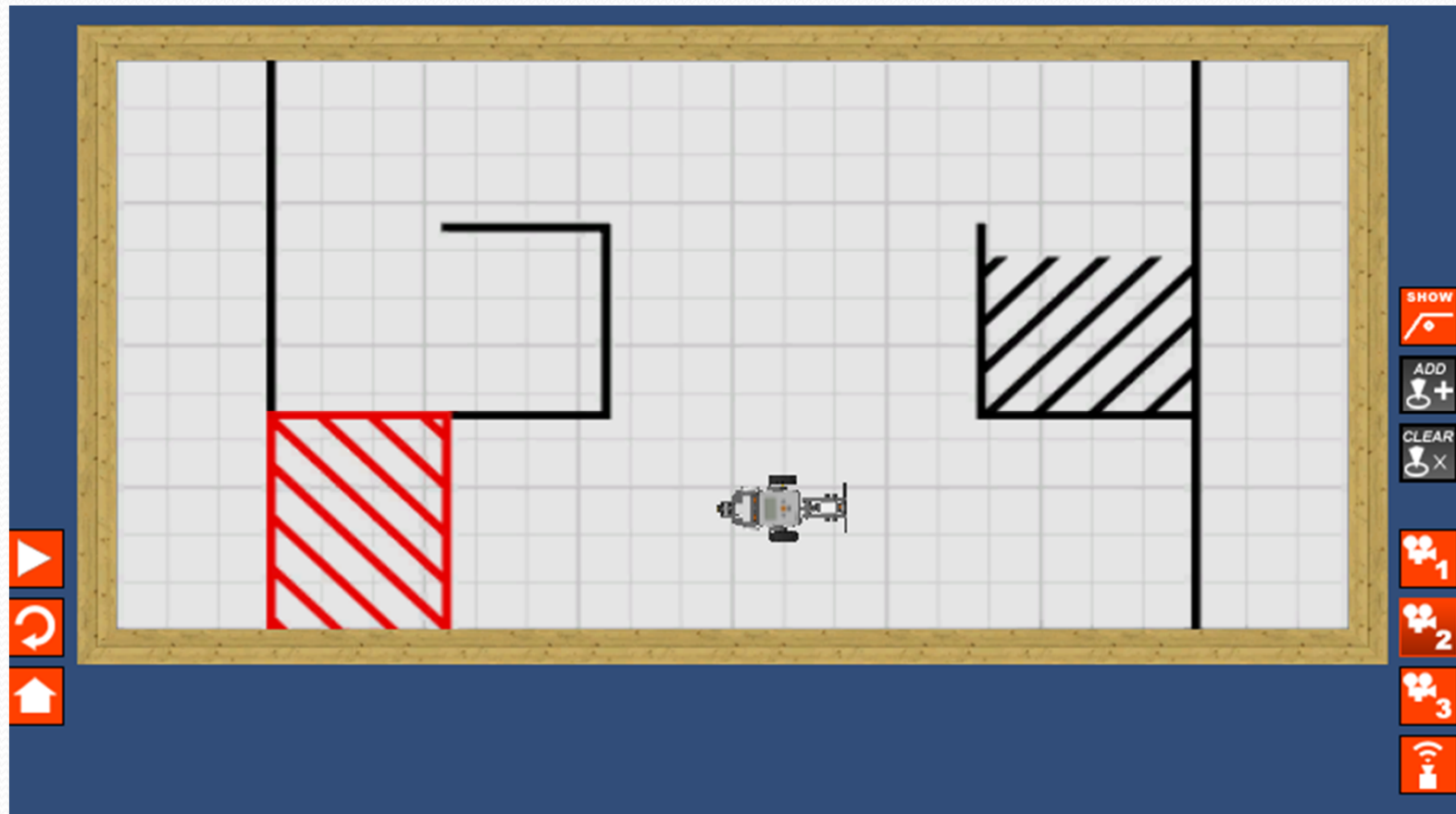
- Aparece el siguiente ambiente una vez pinchando en

START ACTIVITY ▶



RobotC

- Pinchar en play y se ejecuta el programa en el robot virtual



RobotC

- Hay que modificar el código para que el Robot entre el el parking!!

```
task main()
{
    //wait1Msec(2000);          //

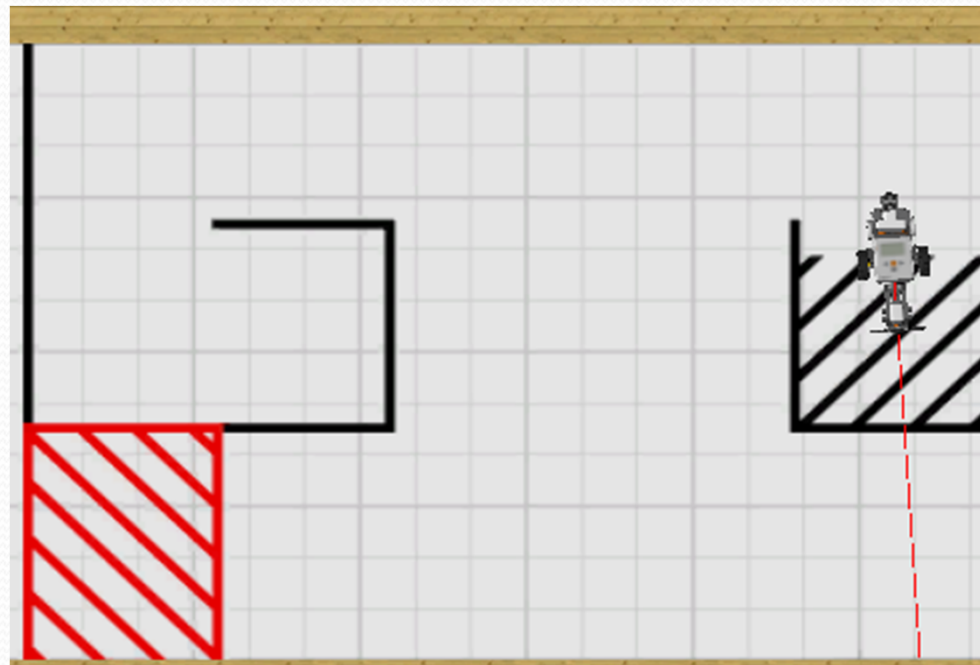
    // Move forward at full power fo
    motor[rightMotor] = 100;    //
    motor[leftMotor]  = 100;    //
    wait1Msec(3000);          //

    //Giro a la izquierda
    motor[rightMotor] = 50;
    motor[leftMotor]  = -50;
    wait1Msec(900);

    //Hacia adelante 2 segundos
    motor[rightMotor] = 100;
    motor[leftMotor]  = 100;
    wait1Msec(2500);

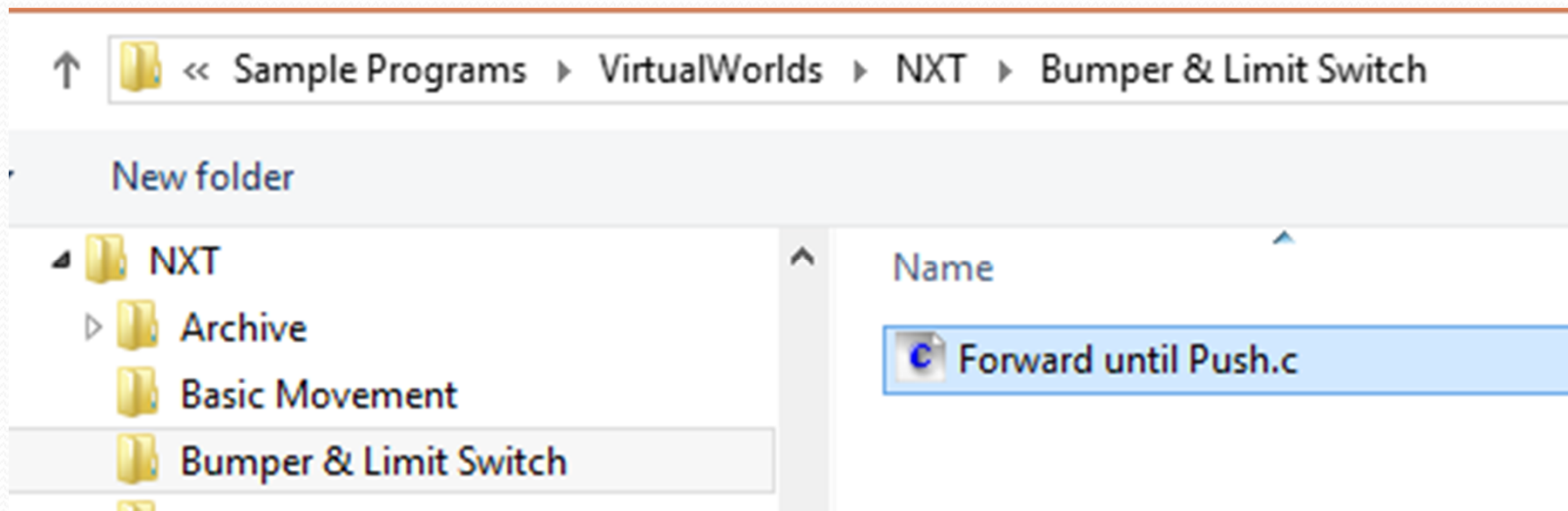
    //Primer Giro a la derecha
    motor[rightMotor] = -50;
    motor[leftMotor]  = 50;
    wait1Msec(900);

    //Hacia adelante 2.5 segundos
    motor[rightMotor] = 100;
    motor[leftMotor]  = 100;
    wait1Msec(2500);
}
```



RobotC

- Vamos a usar el sensor de contacto!!
- Voy a usar otro ejemplo ya hecho tal y como se ve en la siguiente figura:



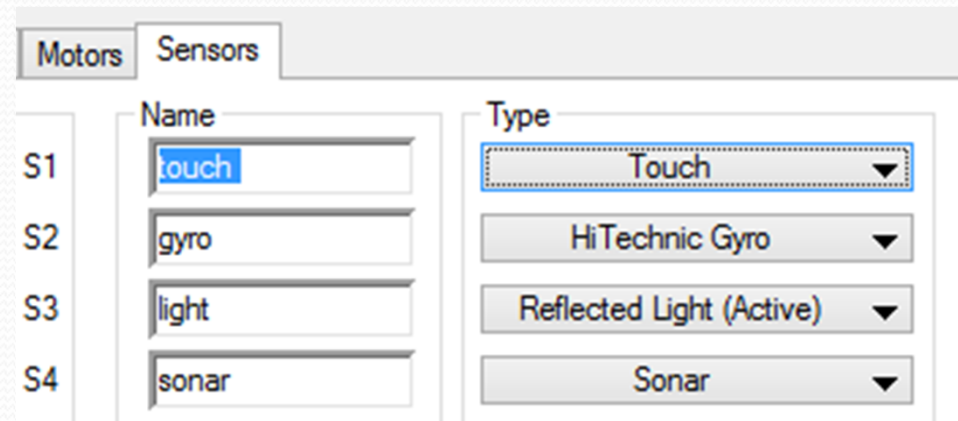
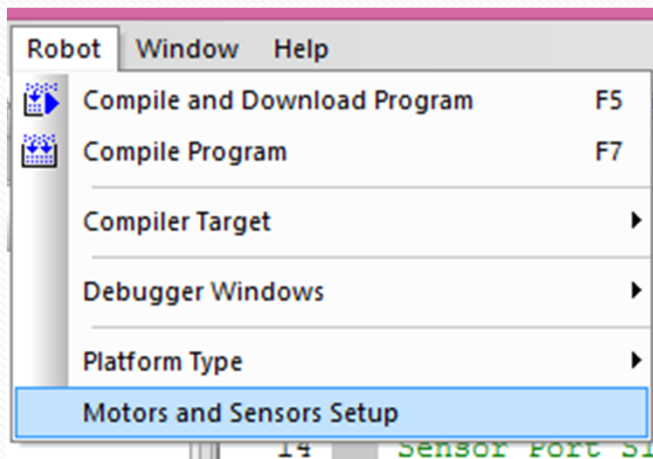
RobotC

- Se ve una estructura de while: que haga lo que está dentro de {} mientras sea verdad lo que tiene entre ():

```
//+++++  
task main()  
{  
    //wait1Msec(2000);  
  
    while(SensorValue(touch) == 0)    // I  
    {  
        motor[rightMotor] = 50;  
        motor[leftMotor] = 50;  
    }  
}  
//+++++
```

RobotC

- Los sensores están definidos en alguna parte de la configuración:



RobotC

- Modifico el código de la siguiente manera:
- Uso el Robo 500 2 en la parte de SENSING

```
task main()
{
    //wait1Msec(2000);

    while(SensorValue(touch) == 0)
    {
        motor[rightMotor] = 50;
        motor[leftMotor] = 50;
    }

    motor[rightMotor] = -50;
    motor[leftMotor] = -50;
    wait1Msec(200);

    motor[rightMotor] = -50;
    motor[leftMotor] = 50;
    wait1Msec(800);
}
```

Robo 500 2

Program the robot to complete two laps around the given square course. The robot must use the touch sensor to accomplish this task.



Specification Document



End of Class 13