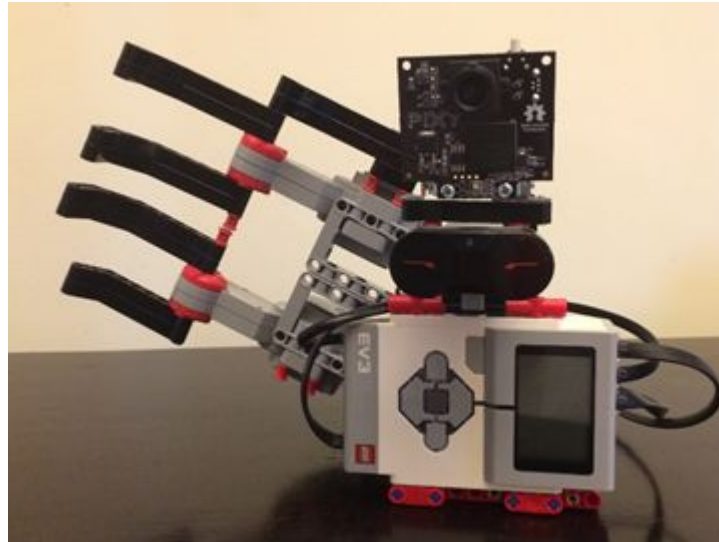
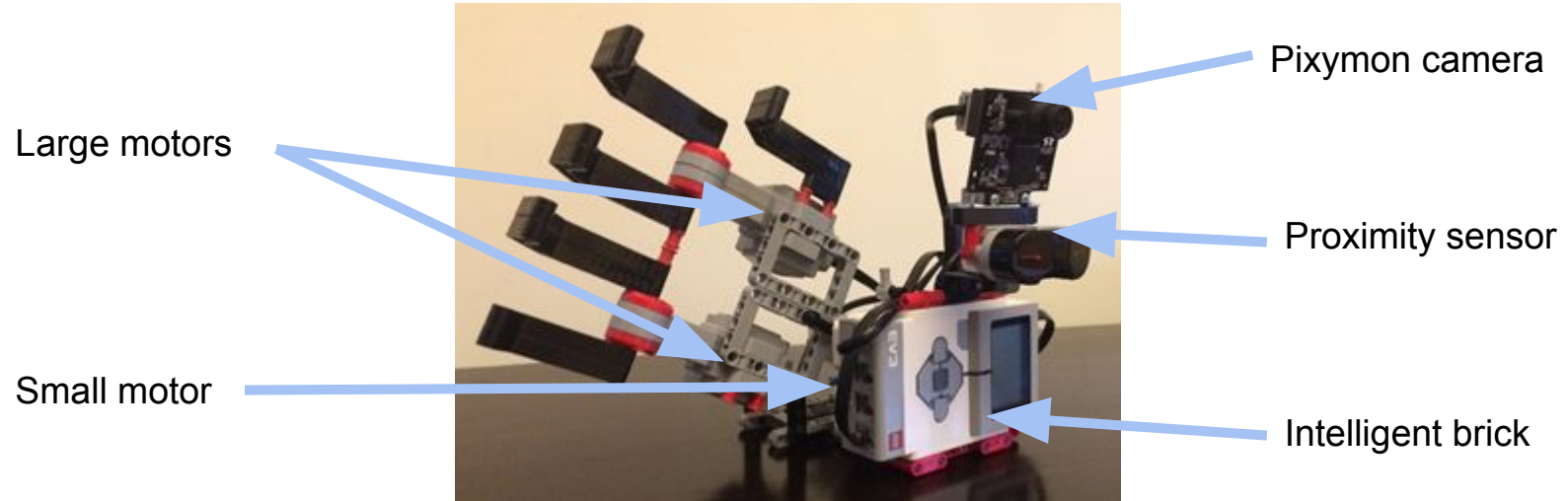


Janken Robot

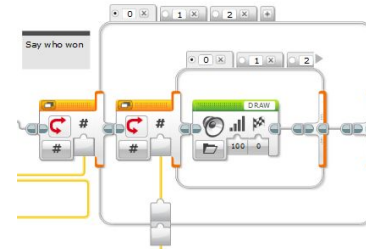
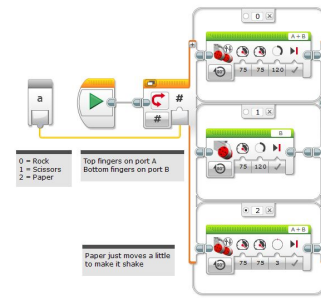
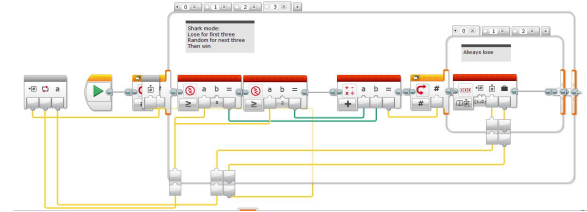
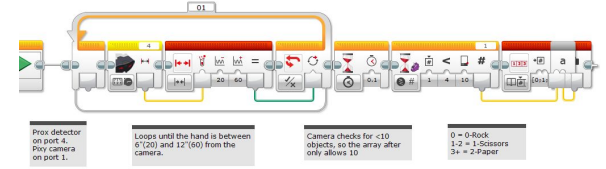
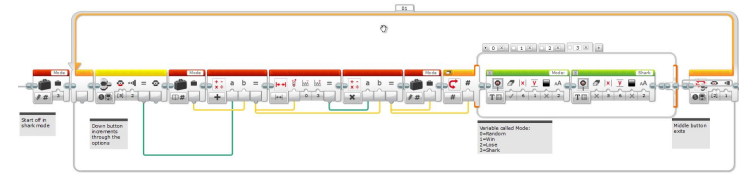
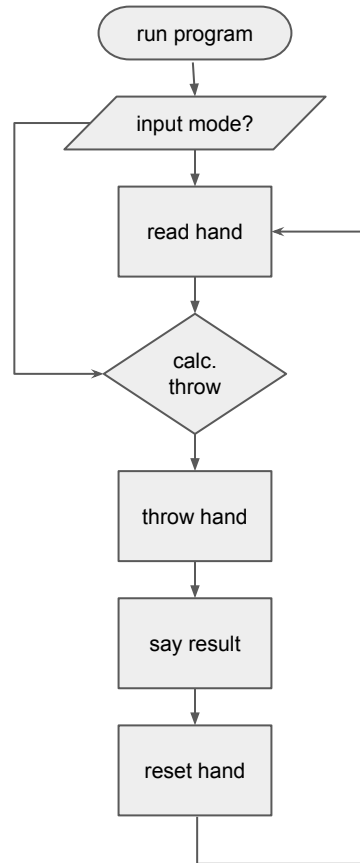


Rebecca Ashmore

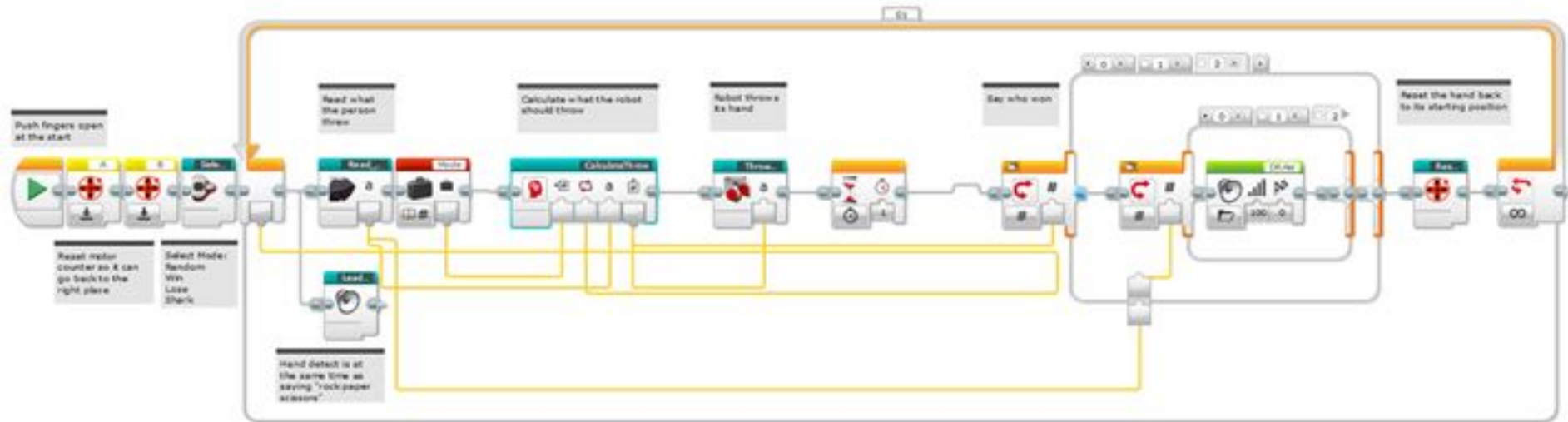
Overview



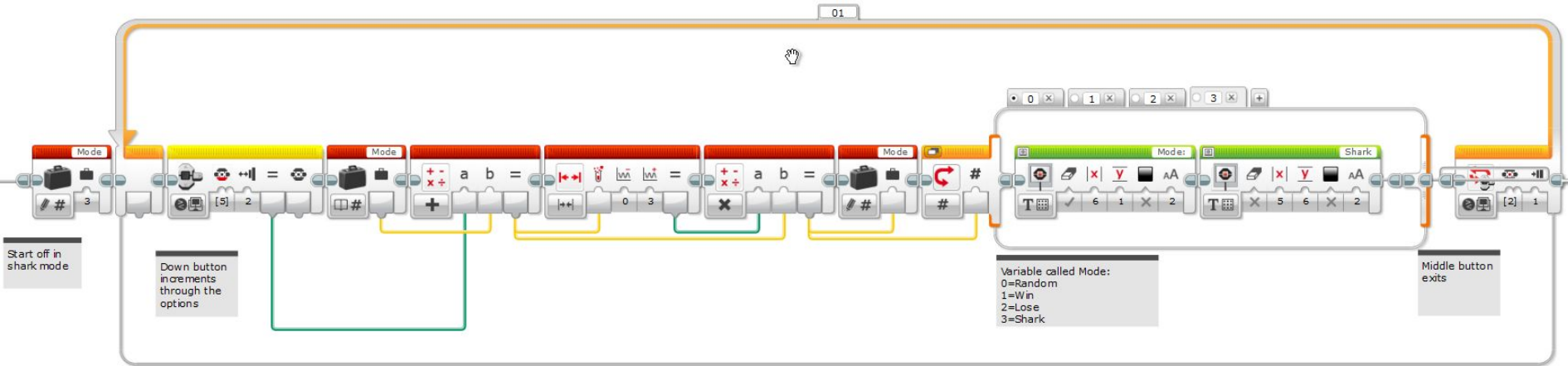
Main program - flowchart



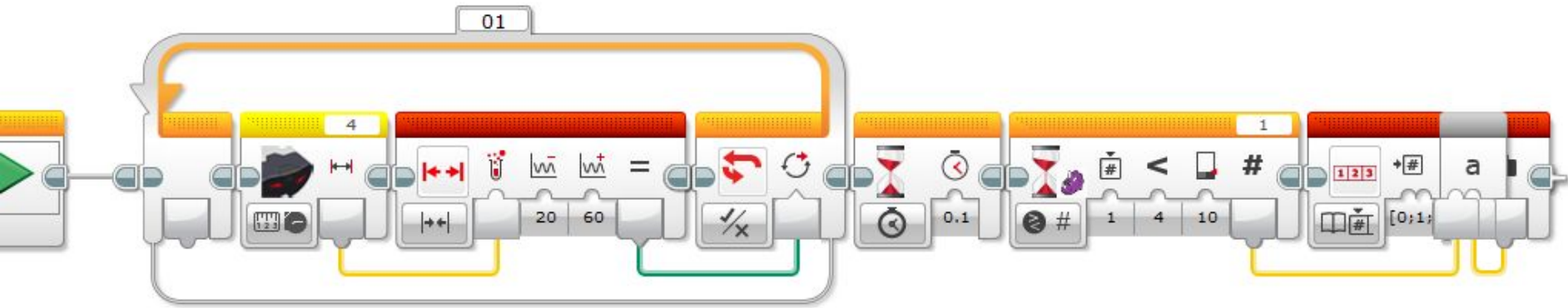
Main program - LabView



Set Mode module



Read hand module



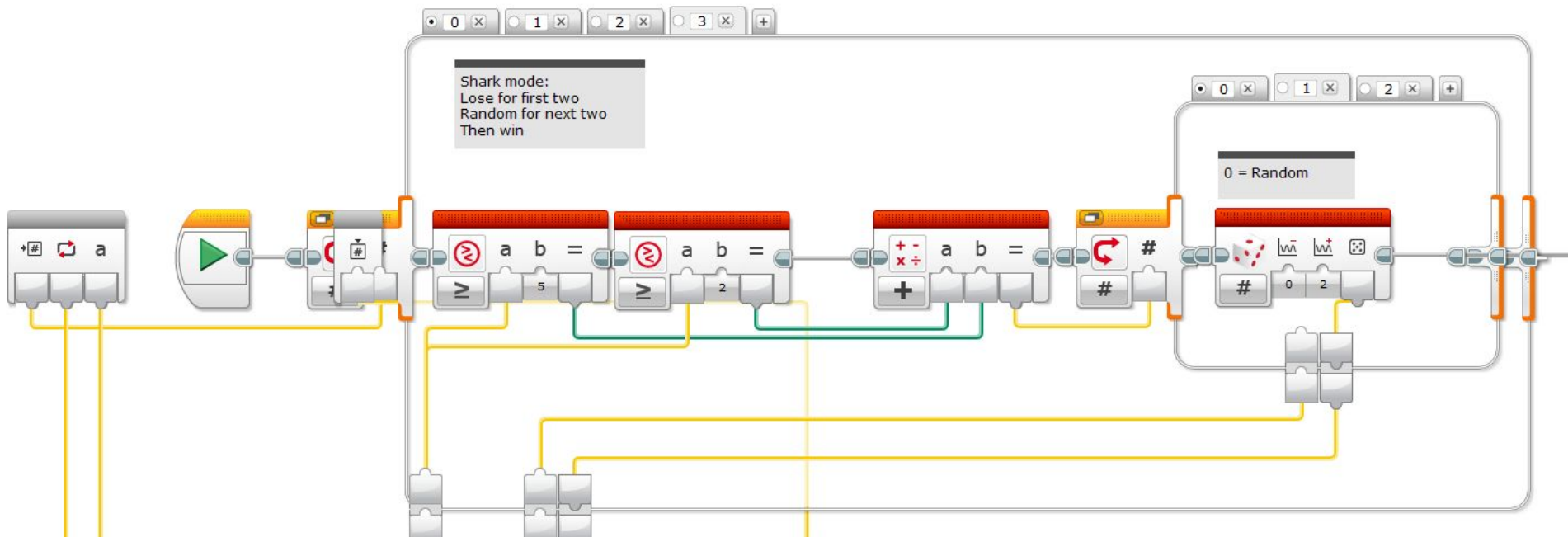
Prox detector
on port 4.
Pixy camera
on port 1.

Loops until the hand is between
6"(20) and 12"(60) from the
camera.

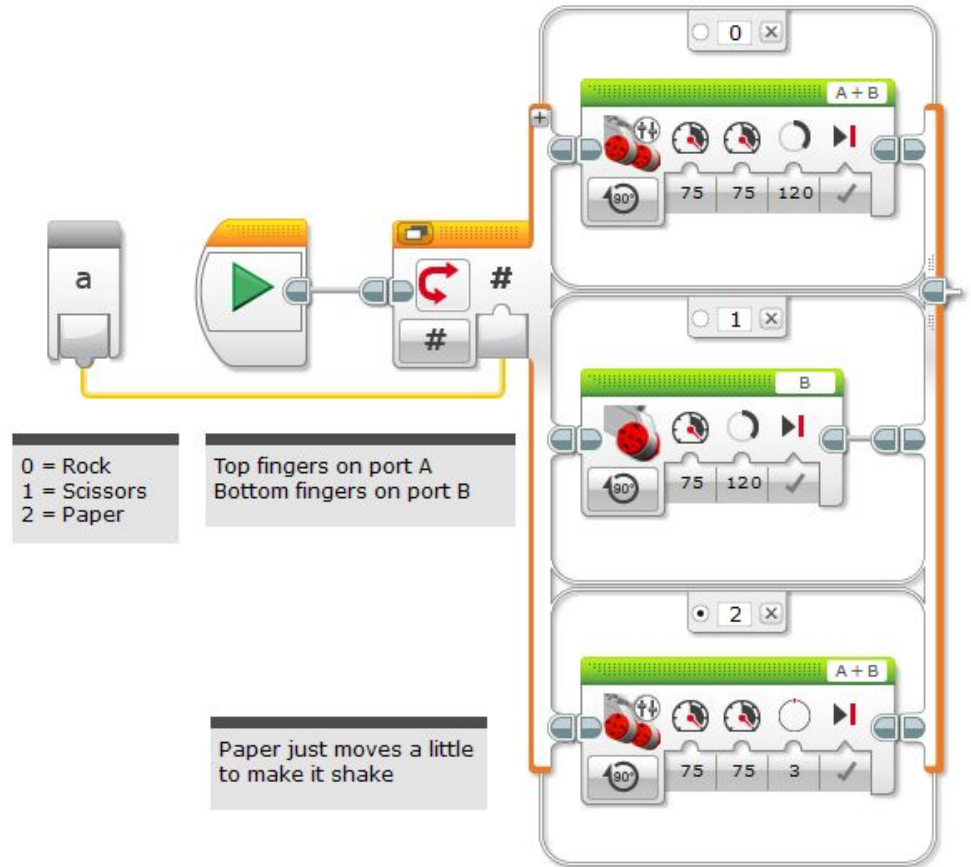
Camera checks for <10
objects, so the array after
only allows 10

0 = 0-Rock
1-2 = 1-Scissors
3+ = 2-Paper

Brain module

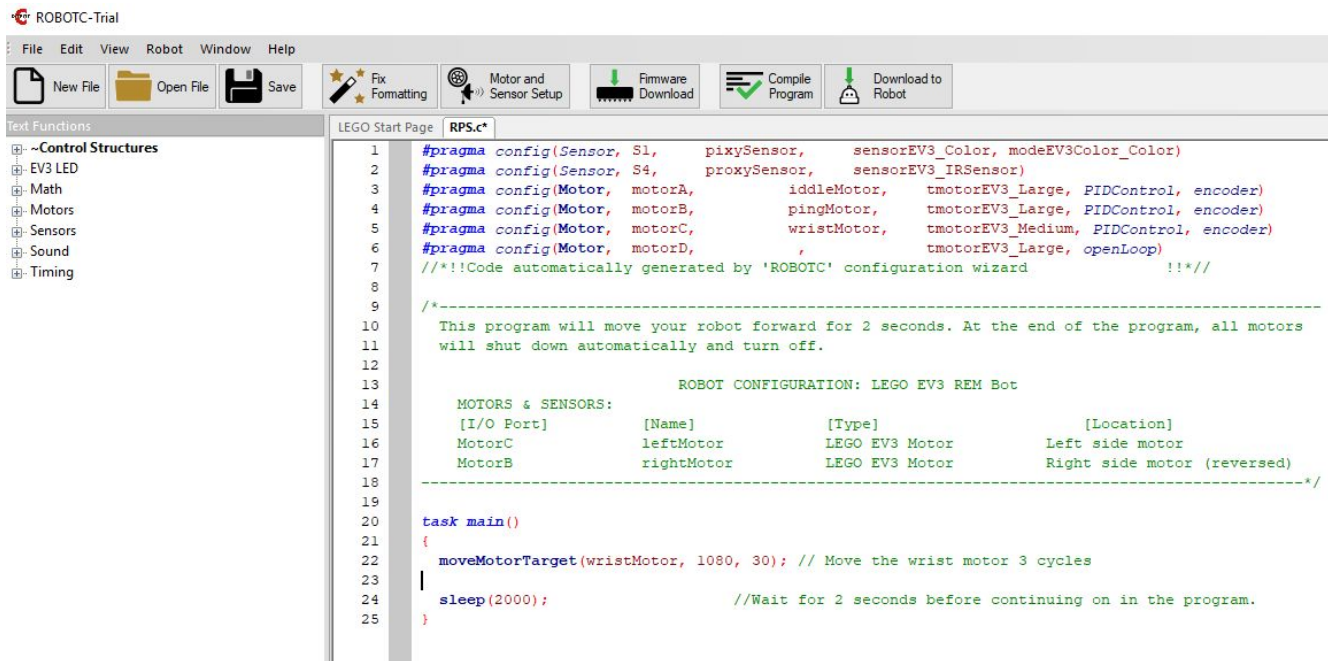


Throw hand module



RobotC

- RobotC is available on Mindstorms, Vex, Arduino
- Possible to re-write a simple version of this RPS program in RobotC



The screenshot displays the RobotC IDE interface. The top menu bar includes File, Edit, View, Robot, Window, and Help. Below the menu is a toolbar with icons for New File, Open File, Save, Fix Formatting, Motor and Sensor Setup, Firmware Download, Compile Program, and Download to Robot. The left sidebar shows a 'Text Functions' tree with categories like ~Control Structures, EV3 LED, Math, Motors, Sensors, Sound, and Timing. The main editor window is titled 'RPS.c' and contains the following code:

```
1  #pragma config(Sensor, S1,  pixySensor,  sensorEV3_Color, modeEV3Color_Color)
2  #pragma config(Sensor, S4,  proxySensor,  sensorEV3_IRSensor)
3  #pragma config(Motor,  motorA,  iddleMotor,  tmotorEV3_Large, PIDControl, encoder)
4  #pragma config(Motor,  motorB,  pingMotor,  tmotorEV3_Large, PIDControl, encoder)
5  #pragma config(Motor,  motorC,  wristMotor,  tmotorEV3_Medium, PIDControl, encoder)
6  #pragma config(Motor,  motorD,  ,  tmotorEV3_Large, openLoop)
7  /**!!Code automatically generated by 'ROBOTC' configuration wizard  !!**/
8
9
10 /*-----
11  This program will move your robot forward for 2 seconds. At the end of the program, all motors
12  will shut down automatically and turn off.
13
14  ROBOT CONFIGURATION: LEGO EV3 REM Bot
15
16  MOTORS & SENSORS:
17  [I/O Port]  [Name]  [Type]  [Location]
18  MotorC      leftMotor  LEGO EV3 Motor  Left side motor
19  MotorB      rightMotor  LEGO EV3 Motor  Right side motor (reversed)
20  -----*/
21
22 task main()
23 {
24   moveMotorTarget(wristMotor, 1080, 30); // Move the wrist motor 3 cycles
25   sleep(2000);                          //Wait for 2 seconds before continuing on in the program.
26 }
```

Robot in action

