

Мастер-класс

Практическое программирование

Головина Олеся Рабадановна



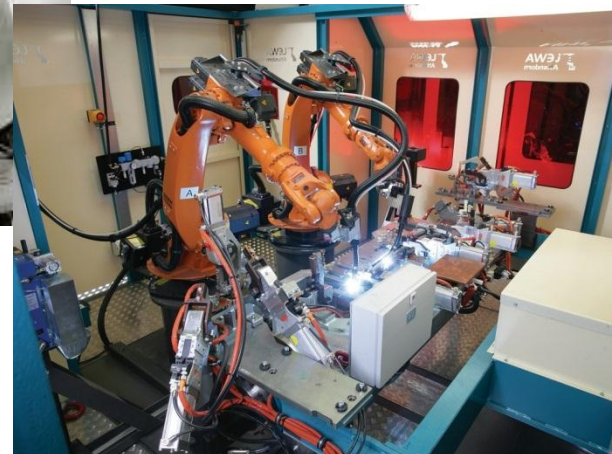
Оглавление

- * робототехнические системы сегодня __ 3-4
- * знакомство с «RobotC» __ 5-9
- * алгоритмические конструкции __ 10
- * основы программирования в «RobotC» __ 11-13
- * органы чувств робота __ 14
- * первая программа в «RobotC» __ 15-29
- * работа над ошибками __ 30

Робототехнические системы сегодня

- * **гибкие автоматизированные производственные системы** - сочетают автоматизацию решения технологических, экономических и административно-управленческих задач;
- * **бионическое протезирование** – выполняет функции утраченных конечностей , органов чувств;
- * **групповая робототехника** - «коллективный разум»;
- * **системы искусственного интеллекта** - робот Элби;
- * **медицинские роботы** - высокоэффективные манипуляторы и инструменты для проведения операций;
- * **военные и спасательные роботы** - экзоскелеты и другие «расширяющие» возможности человека устройства;
- * **домашние роботы** решающие повседневные бытовые задачи;
- * **развлекательные роботы**, реализующие системы виртуальной реальности.

Робототехнические системы сегодня



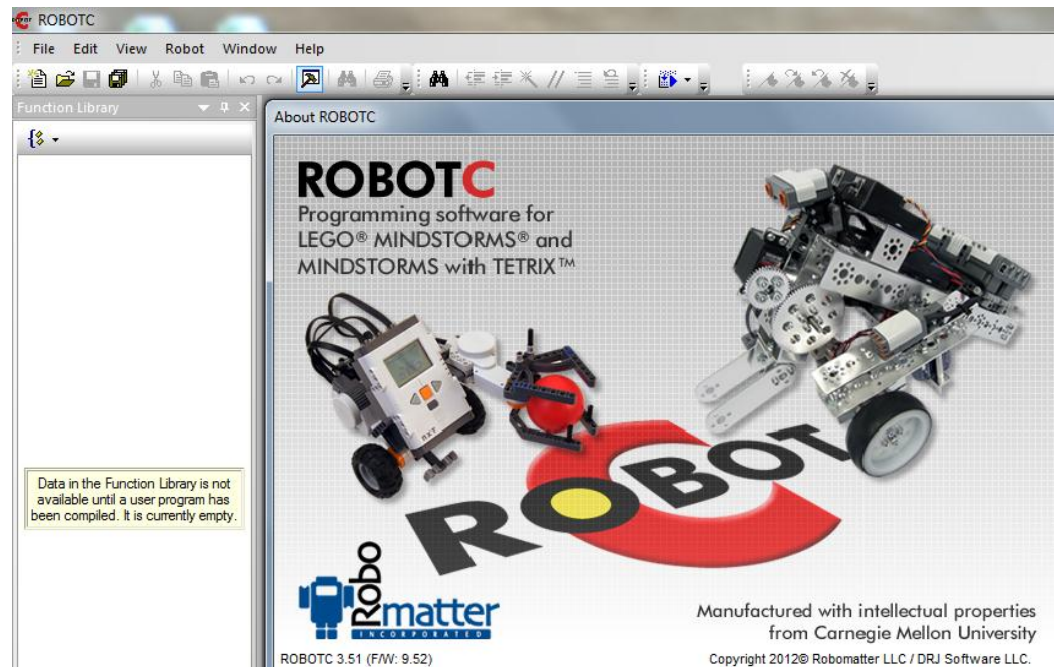
Знакомство со средой программирования RobotC

- * RobotC – среда программирования, основанная на языке Си;
- * является результатом сотрудничества Carnegie Mellon ® и Lego®;
- * создана для развития математических, информационных и технологических компетентностей у обучающихся в сфере образовательной робототехники.

Знакомство со средой программирования RobotC.

Установка

Демонстрационная версия (до 10 дней) на:
<http://www.robotc.net/download/lego/>



Знакомство со средой программирования RobotC.

Прошивка контроллера

Загрузка любой программы в память робота осуществляется:

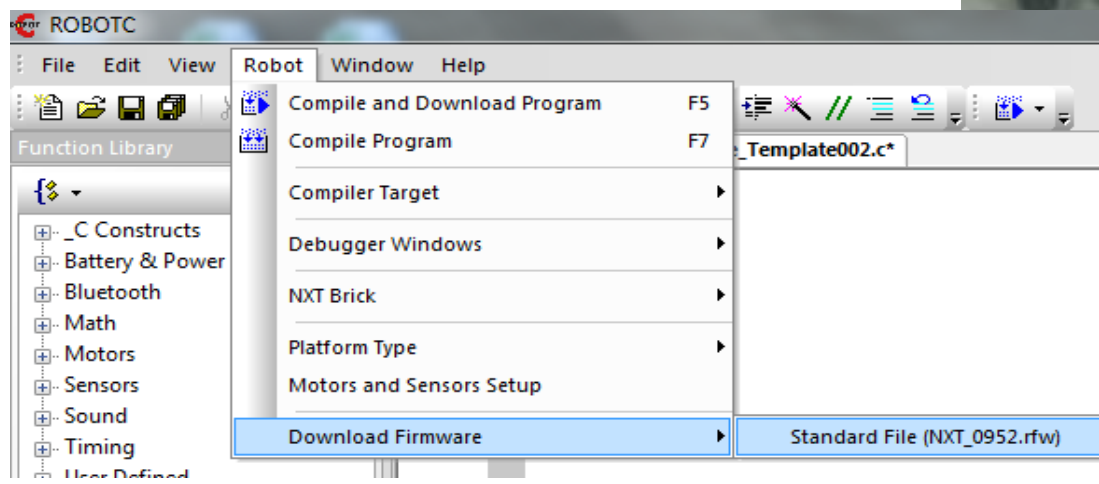
- * посредством USB-кабеля
- * при включенном контроллере
- * «прошить кирпич» - пункт меню

Robot – Download Firmware

1



2

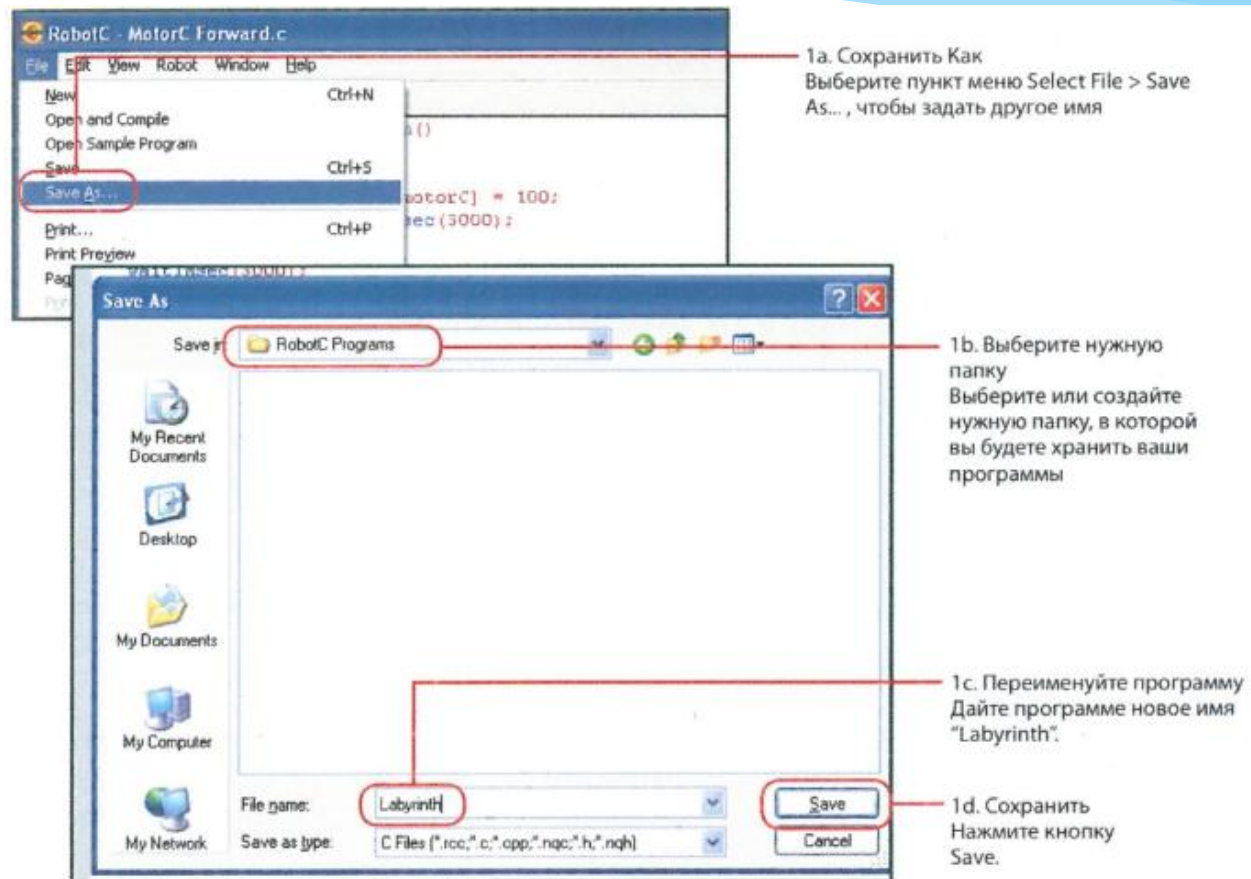


Знакомство со средой программирования RobotC.

Интерфейс. Стандартные

В меню программы вы найдете стандартные пункты:

- * File
- * Edit
- * View

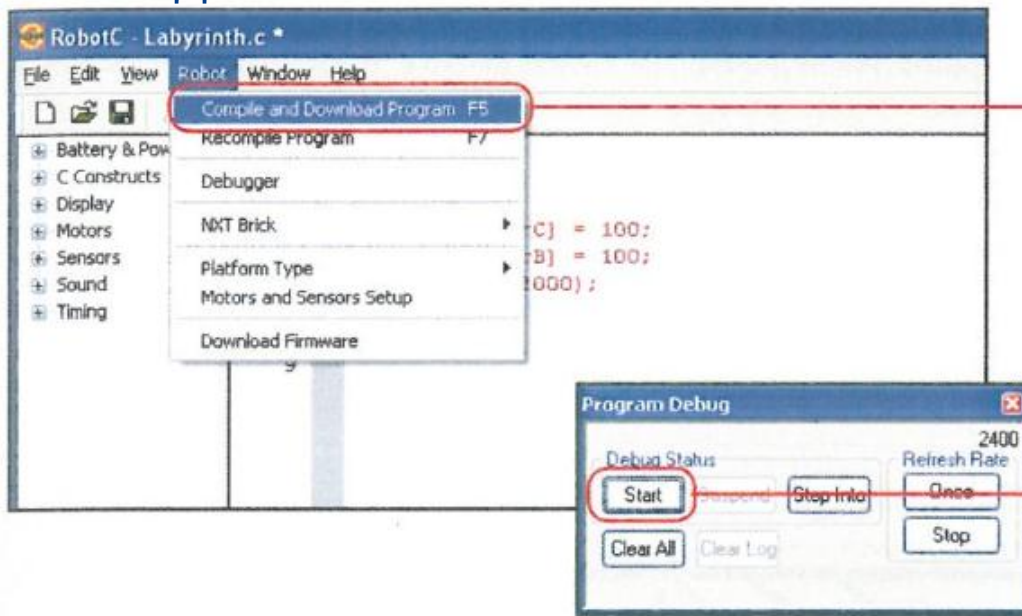


Знакомство со средой программирования RobotC.

Интерфейс. Специальные

В меню программы есть специальные пункты:

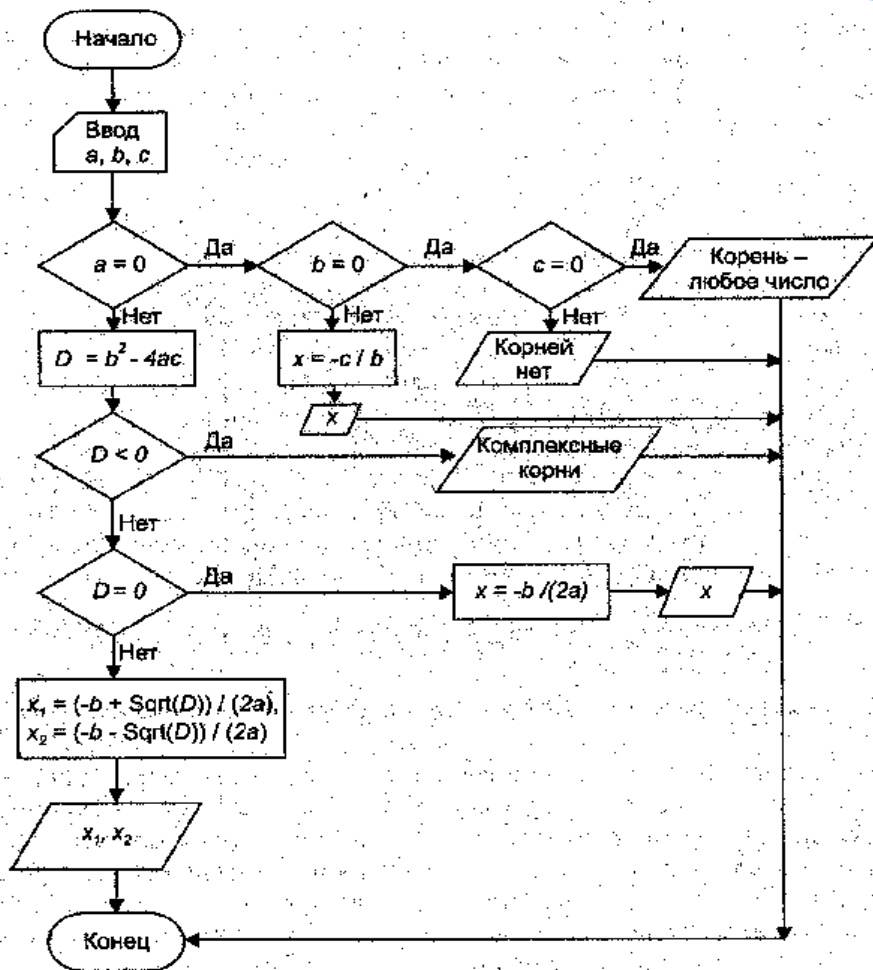
- * Robot – предназначен для работы с роботом
- * Window – позволяет выбрать вид панели инструментов и команд слева



2a. Загрузите программу
Выберите пункт меню Robot >
Compile and Download Program
что бы загрузить программу

2b. Нажмите кнопку Start
Нажмите кнопку Start в
появившемся окне, что бы
запустить программу .

Представление алгоритмических конструкций



«идеальный алгоритм»

Основы программирования в RobotC. Правила - I

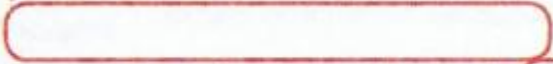
- * перед выполнением программу необходимо откомпилировать: меню Robot – Compile program
- * язык чувствителен к регистру
- * ключевые слова и команды подсвечиваются
- * операторные скобки – фигурные скобки
- * в конце каждой команды ставится знак «;»

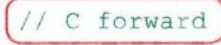
```
1 task main()  
2 {  
3  
4     motor[motorC] = 0;  
5     wait1Msec(3000);  
6  
7 }
```

Точка с запятой
Она обозначает конец каждой
команды в ROBOTC.

Основы программирования в RobotC. Правила - II

- * пробелы и пустые строки не влияют на работу программы
- * в квадратных скобках [] указываются идентификаторы
- * в круглых скобках () указываются значения или выражения
- * все скобки используются попарно
- * для комментариев используются знаки: // и /* ... */

```
1 task main()  
2 {  
3       
4     motor[motorC] = 0;   
5     wait1Msec(3000);   
6       
7 }
```

```
1 task main()  
2 {  
3  
4     motor[motorC] = 100;   
5     wait1Msec(3000);  
6  
7 }
```

Комментарии: // Одна строка
Любая часть текста, которая
следует за // (двойным
слешем), считается
комментарием

```
1 task main()  
2 { /* this is all  
3     part  
4     of the  
5     same comment */   
6     motor[motorC] = 100;  
7     wait1Msec(3000);
```

Комментарии: /* Многострочные*/
Комментарии в ROBOTC можно делать
используя пару символов /* и */
Этот тип комментариев позволяет
использовать несколько строк

Основы программирования в RobotC. Правила - III

- * каждая программа содержит специальную часть, которая называется главной задачей: `task main()` - это та часть программы, которую и будет выполнять робот

```
1 task main()  
2 {  
3  
4     motor[motorC] = 100;  
5     wait1Msec(3000);  
6  
7 }
```

`task main()`
Эта строка создает главную задачу, которая содержит код, который робот должен выполнить.

`task main()` обозначает начало структуры

{ тело программы }
Скобки { и } ограничивают код главной задачи. Строки между скобками говорят программе, какие именно команды содержатся в главной задаче

Органы чувств робота.

Датчики

Датчики, используемые для ориентирования в пространстве:

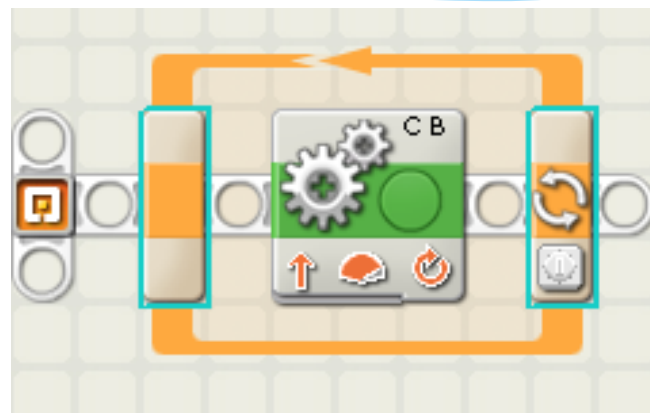
- * цвета/света
- * УЗД
- * ИК



Первая программа в RobotC.

Сравнение с программой в нативной среде

```
1 task main()  
2 {  
3  
4     motor[motorC] = 100;  
5     motor[motorB] = 100;  
6     wait1Msec(3000);  
7  
8 }
```



Задание 1. Уменьшите скорость движения робота.
Загрузите программу, выполните.

Задание 2. Увеличьте время движения робота до 6 с.
Загрузите программу, выполните.

Первая программа в RobotC.

Движение вперед

```
1 task main()  
2 {  
3  
4     motor[motorC] = 100;  
5     motor[motorB] = 100;  
6     wait1Msec(3000);  
7  
8 }
```

Задание 1. Уменьшите скорость движения робота.
Загрузите программу, выполните.

Задание 2. Увеличьте время движения робота до 6 с.
Загрузите программу, выполните.

Первая программа в RobotC.

Маневрирование

Команды моторов	Движение в результате
<pre>motor[motorC]=100; motor[motorB]=100;</pre>	 A diagram of a three-wheeled robot with a red arrow pointing straight up from its front, indicating forward movement.
<pre>motor[motorC]=50; motor[motorB]=50;</pre>	 A diagram of a three-wheeled robot with a small red arrow pointing straight up from its front, indicating slow forward movement.
<pre>motor[motorC]=-100; motor[motorB]=-100;</pre>	 A diagram of a three-wheeled robot with a red arrow pointing straight down from its front, indicating backward movement.
<pre>motor[motorC]=0; motor[motorB]=0;</pre>	 A diagram of a three-wheeled robot with a red dot above it, indicating it has stopped.
<pre>motor[motorC]=100; motor[motorB]=-100;</pre>	 A diagram of a three-wheeled robot with a red circular arrow pointing clockwise around it, indicating a right turn.
<pre>motor[motorC]=100; motor[motorB]=0;</pre>	 A diagram of a three-wheeled robot with a red circular arrow pointing counter-clockwise around it, indicating a left turn.

Задание 3.
Запрограммируйте
вращение робота в правую
сторону в течение 6 с.
Загрузите программу,
выполните.

Задание 4. Выполните
разворот робота вокруг
собственной оси на 180°.
Загрузите программу,
выполните.

Первая программа в RobotC.

Маневрирование. Особенности

Используя команду `motor[]` мы устанавливаем мощность. Однако, ввиду различных производственных факторов, на самом деле двигатели вращаются с разной скоростью. Это значит, даже если мы подаем на моторы одинаковую мощность, подаваемая энергия на колеса все равно может отличаться. Можно подобрать значения для каждого двигателя в ручную. Но есть и другой, более эффективный способ.

Первая программа в RobotC.

Маневрирование. PID-регулирование

RobotC включает в себя специальный PID-алгоритм.

PID-алгоритм это обработка:

«П» - пропорциональной,

«И» - интегральной,

«Д» - дифференциальной

составляющей сигнала обратной связи.

PID-контроль позволяет рассчитывать мощность, которую необходимо подать на мотор, чтобы колесо вращалось с заданной скоростью, на основе датчика количества оборотов, встроенного в мотор.

Первая программа в RobotC.

Маневрирование. PID-регулирование



Начальная позиция (t=0)
Из этой позиции колесо начинает вращаться

И чуть позже... (t=0.1s)
Спустя 1/10 долю секунды, колесо уже срезка повернулось. Зная, насколько повернулось колесо за какое время, робот может рассчитать скорость вращения.

Первая программа в RobotC.

Маневрирование. PID-регулирование

```
1  task main()  
2  {  
3      nMotorPIDSpeedCtrl[motorB]=mtrSpeedReg;  
4      nMotorPIDSpeedCtrl[motorC]=mtrSpeedReg;  
5  
6      motor[motorC]=50;  
7      motor[motorB]=50;  
8      wait1Msec(3000);  
9  
10     motor[motorC]=-50;  
11     motor[motorB]=50;  
12     wait1Msec(800);  
13  
14 }
```

Первая программа в RobotC.

Маневрирование. Синхронизация моторов

Синхронизация моторов позволяет «уравновесить» скорость вращения моторов. Для синхронизации используется специальная команда .

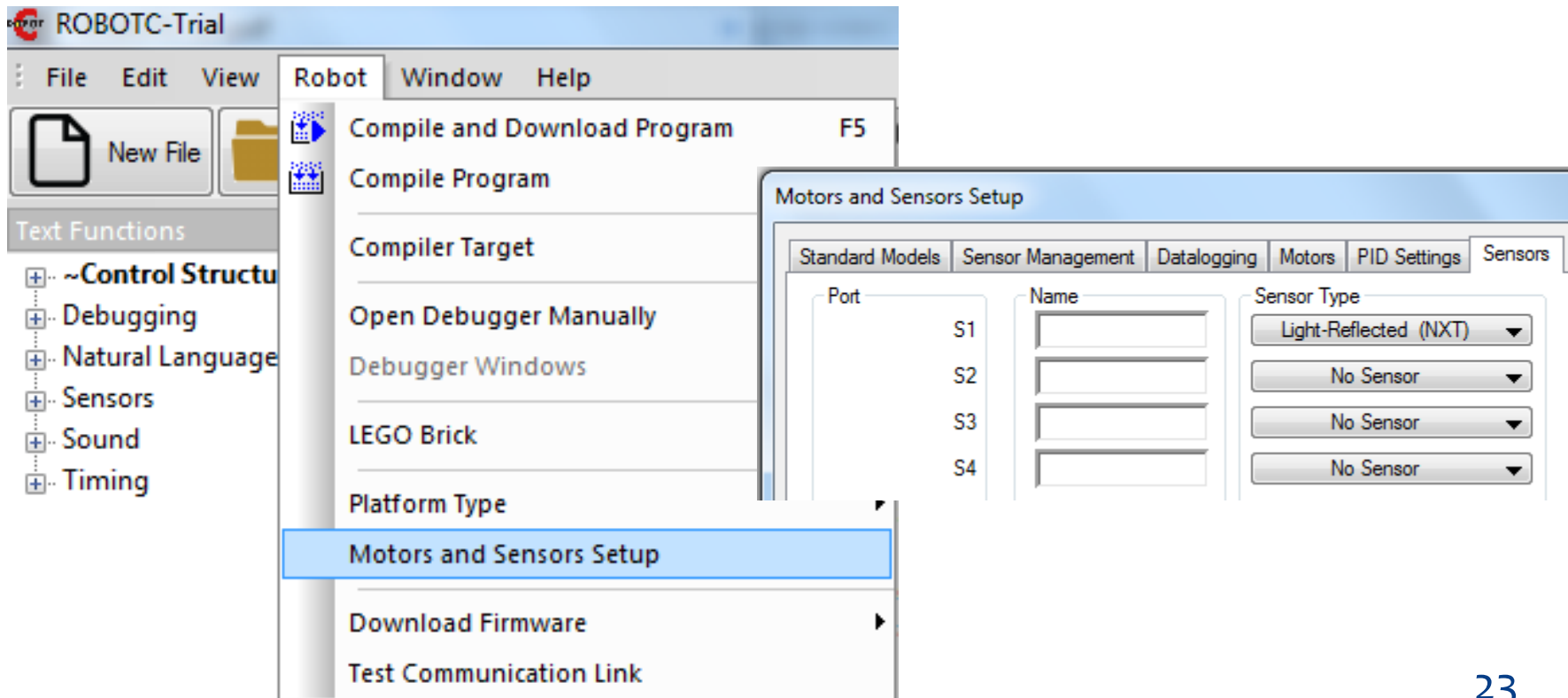
```
1  task main()  
2  {  
3      nSyncedMotors=synchBC;  
4      nSyncedTurnRatio=100;  
5  
6      motor [motorB]=50;  
7      wait1Msec (3000);  
8  
9  }
```

- последовательность важна: первым указывается ведущий мотор

Первая программа в RobotC.

Ориентирование по датчикам. Датчик света

Подключим датчик отраженного света: Robot – Motors and Sensors Setup. Установим нужное значение у датчика S1.



Первая программа в RobotC.

Ориентирование по датчикам. Датчик света

В программе появляется строка, где необходимо только указать название переменной, которая хранит показания датчика

```
1  #pragma config(Sensor, S1, liteSensor,  sensorLightActive)
2
3  task main()
4  {
5      while (SensorValue(liteSensor)>40)
6      {motor[motorC]=50;
7      motor[motorB]=50;
8      }
9      motor[motorC]=0;
10     motor[motorB]=0;
11
12 }
```

В программе появляется строка, где необходимо только указать название переменной, которая хранит показания датчика

Первая программа в RobotC.

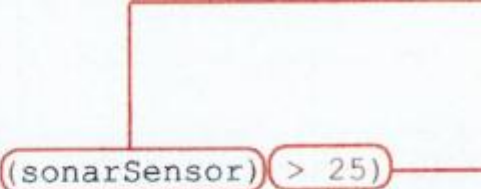
Ориентирование по датчикам. УЗД датчик

Задание 5. Используйте УЗД по образцу предыдущей программы

Первая программа в RobotC.

Ориентирование по датчикам. УЗД датчик

```
1
2 task main()
3 {
4
5     while (SensorValue(sonarSensor) > 25)
6     {
7
8         motor[motorC] = 50;
9         motor[motorB] = 50;
10
11     }
12
13     motor[motorC] = -50;
14     motor[motorB] = -50;
15     wait1Msec(2000);
16
17 }
```



Первая программа в RobotC.

Ориентирование по датчикам. УЗД датчик

Задание 6. Запрограммируйте чтобы робот двигался вперед, а за 30 см до препятствия поворачивал налево. Создайте бесконечный цикл.

Спасибо за внимание!

Напишите, пожалуйста
что нового вы узнали на этом
занятии.

Будем рады видеть вас в числе
студентов нашего центра!