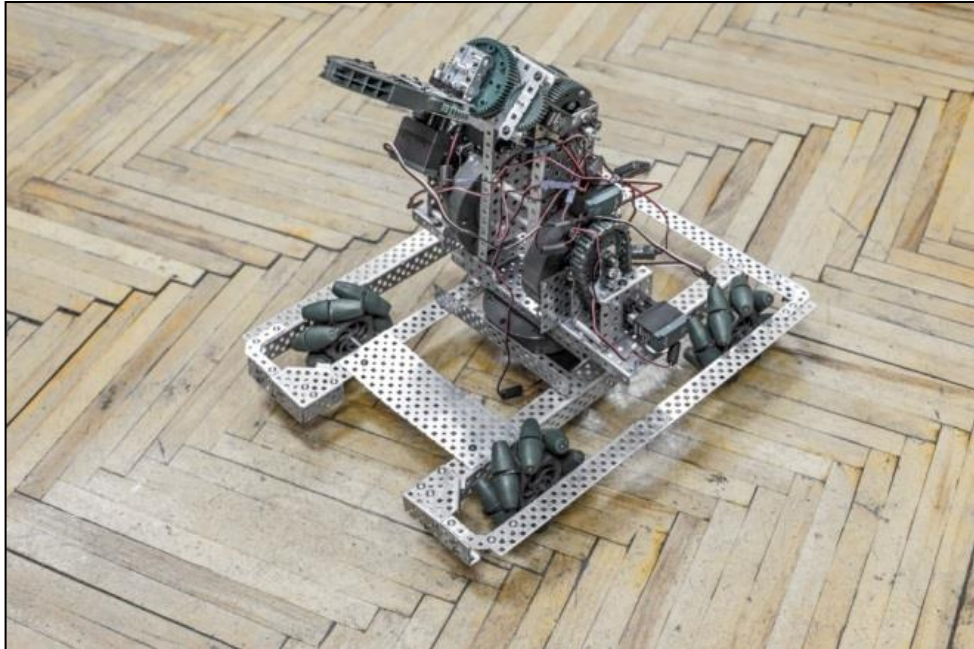


ЛАБОРАТОРНАЯ РАБОТА N 9

«Сборка и программирование макета манипуляционного робота»

Цель работы: сборка и программирование макета манипуляционного робота на подвижной платформе на базе робототехнического конструктора VEX.



1. Теоретическая часть

В настоящее время наибольшее распространение в мире получили промышленные и различные типы мобильных роботов (МР), оснащенные манипуляционным механизмом. В свою очередь, среди МР особое значение приобрели автономные роботы, предназначенные для решения экстремальных задач, в частности, для выявления и уничтожения взрывчатых веществ. Можно отметить разработки зарубежных компаний Andros Mk V A, Wheelbarrow Mk7, RODE, Castor, Hobo и др. В России за короткий период созданы семейства МР «МПК-01», «МПК-25», «Скорпион», «Вездеход-ТМЗ» (рис. 10.1, *а*) и «Варан» (рис. 10.1, *б*), являющиеся результатом совместного сотрудничества специалистов различных областей.



Рис. 10.1. МР для экстремальной робототехники: *а* — «Вездеход-ТМЗ», *б* — «Варан»

«Вездеход-ТМЗ» относится к роботам сверхлегкого класса, основным назначением которых является визуальная и акустическая разведка местности, помещений, транспортных средств, осмотр труднодоступных мест, обнаружение и уничтожение взрывчатых веществ. В его состав

входят: переносной пульт дистанционного управления, колесное транспортное средство, рабочее оборудование – манипулятор, механизмы наведения и гидроразрушитель.

MP «Варан» – это универсальный робот легкого класса, предназначенный для проведения телевизионной разведки объектов и территорий в условиях городской инфраструктуры и пересеченной местности; осмотра и исследования объектов, расположенных в помещениях и на местности, а также автотранспортных средств; уничтожения взрывчатых веществ или их загрузку в специальный контейнер для транспортирования в безопасное место. Он состоит из гусеничного транспортного средства высокой проходимости, рабочего оборудования (манипулятора, разрушителей и пр.), пульта дистанционного управления.

Макеты MP собираются из элементов роботехнического конструктора VEX EDR, компании VEX Robotics (США). Конструкторы этой компании широко используют в лабораториях различных американских университетов (рис. 10.2). В состав набора входят перфорированные металлические детали – профили и пластины, пластиковые элементы передач – зубчатые колеса, шкивы и колеса и т.п. (рис. 10.3, а, б).

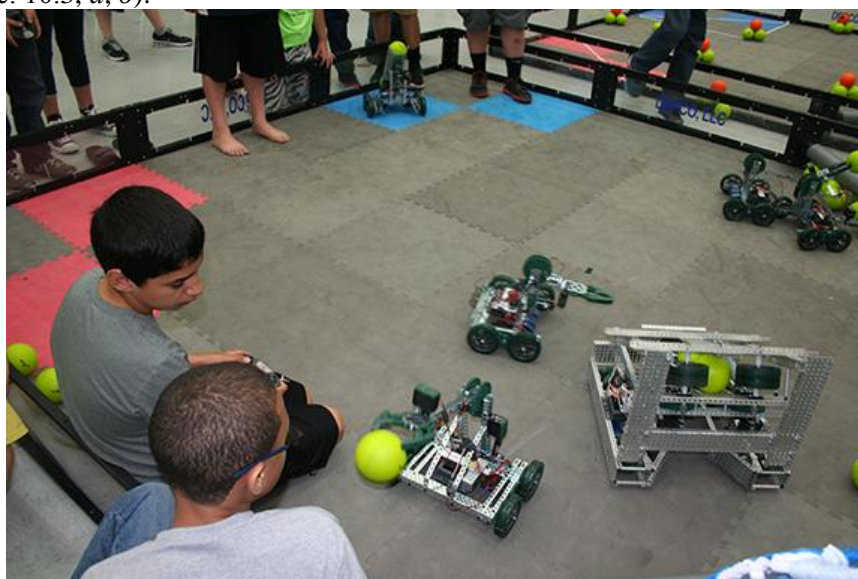


Рис. 10.2. Фрагмент университетских соревнований 2015 г.

Наборы VEX укомплектованы современными микроконтроллерами VEX Cortex (рис. 10.3, в), сервомоторами и разнообразными датчиками (рис. 10.3, з).

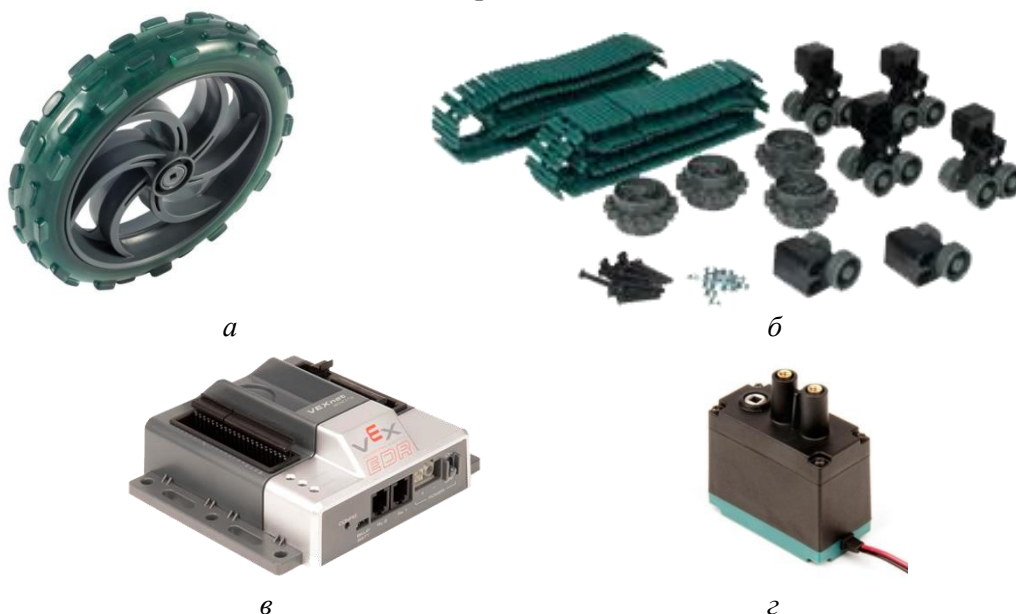


Рис. 10.3. Два типа подвижных платформ, используемых в лабораторной работе: а – колесная, б – гусеничная, в – контроллер VEX Cortex, з – сервопривод Motor269

2. Практическая часть

2.1. Лабораторная установка

В состав лабораторного стенда входят:

- персональный компьютер с операционной системой Windows (версия XP или более поздняя);
- образовательный робототехнический конструктор VEX EDR и ресурсный набор, включающий:
 - приводы Motor 269 (не менее 4шт.);
 - программируемый контроллер VEX Cortex;
 - конструкционные элементы;
- программный комплекс, включающий:
 - ROBOTC for VEX Cortex and PIC.

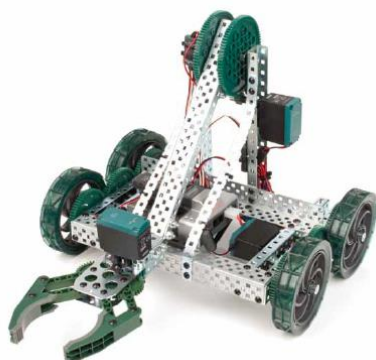
2.2. Порядок выполнения лабораторной работы

Лабораторная работа состоит из двух частей:

1. сборка и программирование подвижной платформы;
2. установка макета манипулятора на подвижную платформу. Программирование робота.

2.2.1. Сборка и программирование подвижной платформы

1. Используя предложенные детали и электрические привода, собрать платформу на основе колесных или гусеничных движителей (рис. 10.4).



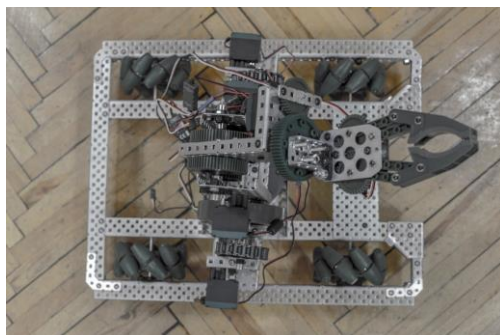
a



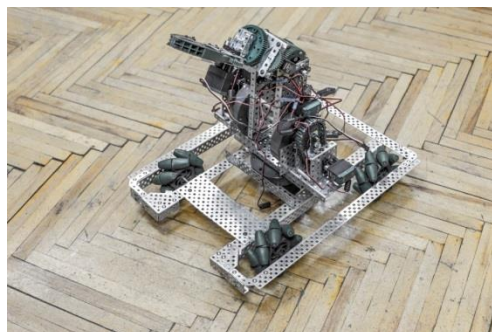
б

Рис. 10.4. Пример платформы на колесном ходу (*a*), собранный МР (*б*).

Примечание 1. Заранее ознакомьтесь с размерами макета манипулятора (рис. 10.5, *в*, *г*). В следующей части лабораторной работы Вам предстоит установка манипулятора на собранную Вами подвижную платформу.



a



б

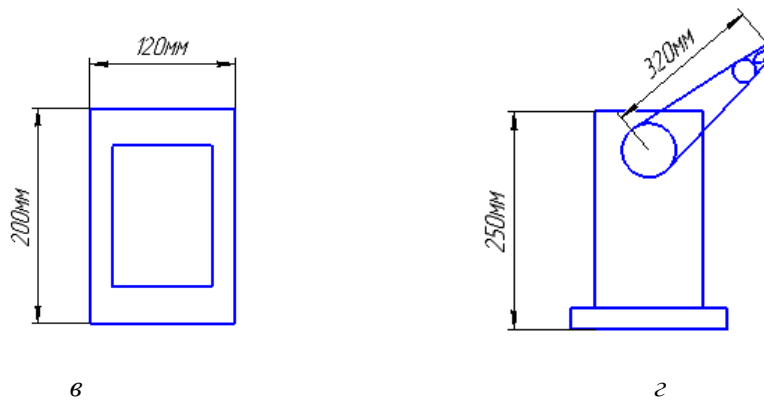


Рис. 10.5. Пример платформы на колесном ходу (а), собранный МР (б), размеры платформы (в) и манипулятора (г)

2. Установите на робота предложенный набор ультразвуковых датчиков Ultrasonic Range Finder компании VEX (рис. 10.6);



Рис. 10.6. Датчик Ultrasonic Range Finder

3. Запрограммируйте робота для езды из точки А в точку В и измерьте время его следования по маршруту (рис. 10.7, а и 10.7, б):
 - а. используя один датчик;
 - б. используя два датчика.

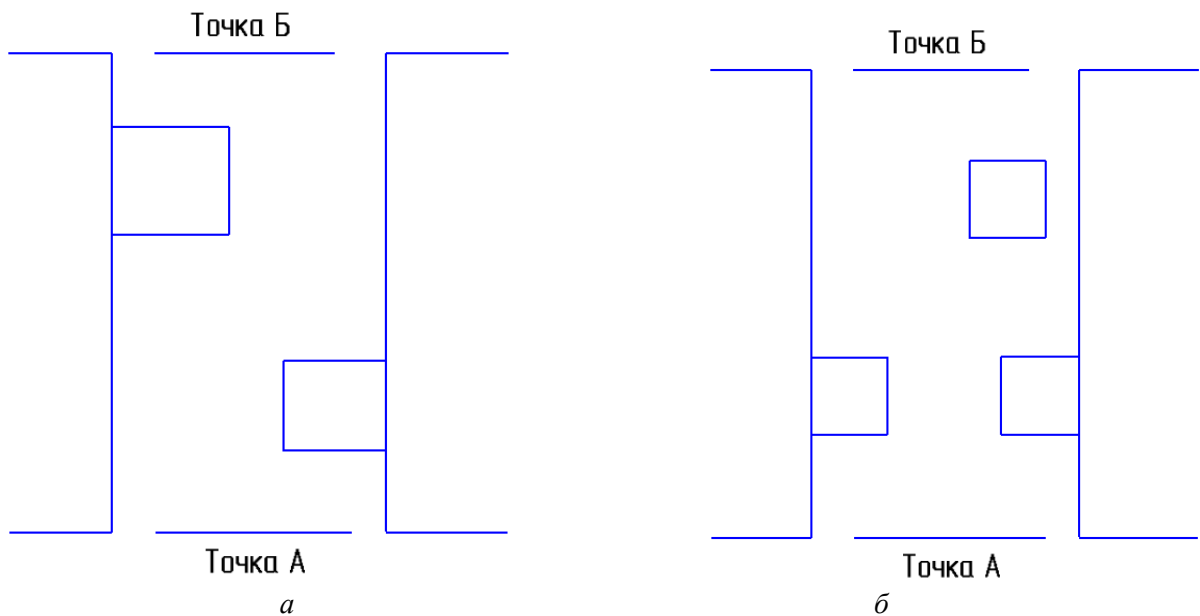


Рис. 10.7. Маршруты следования: а – №1, б – №2

Примечание 2. Маршрут следования определяется преподавателем в начале лабораторной работы.

Примечание 3. Описание среды программирования, а также примеры программ на языке RobotC находятся в приложении к лабораторной работе.

4. Заполните следующую таблицу:

Табл. 1. Время следования робота по маршруту

№	Число датчиков	Время следования по маршруту, с
1	1	
2	2	

- Сравните работу робота при езде с разным количеством датчиков (сравнение производится на основе данных, полученных в пункте 3).
- Сделайте выводы об эффективности работы робота в зависимости от числа датчиков.

2.2.2. Установка макета манипулятора на подвижную платформу. Программирование робота

- Закрепите макет манипулятора на подвижной платформе, собранной в первой части настоящей лабораторной работы;
- Напишите программу работы манипулятора в автономном режиме на неподвижной платформе. Программа должна работать по следующему алгоритму:
 - включение робота;
 - поворот из текущего положения к объекту;
 - захват объекта;
 - перенос объекта в конечное положение;
 - постановка объекта в конечную точку.

Положение объекта в пространстве показано на рис. 10.8.

Примечание к пункту 2. Выполнение этого пункта лабораторной работы определяется преподавателем в начале лабораторной работы.

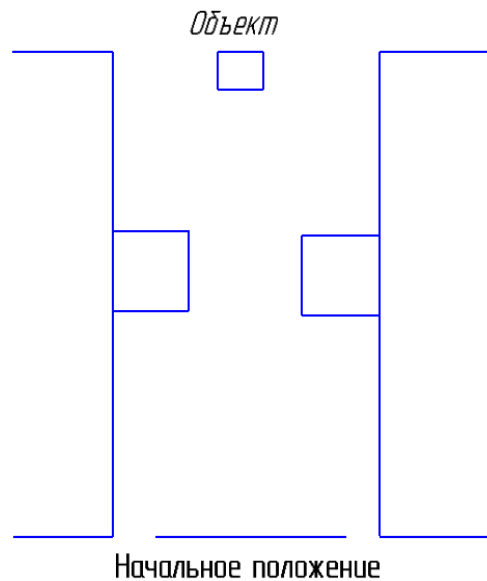


Рис. 10.8. Положение объекта

- Напишите программу работы манипулятора от пульта на подвижной платформе. Программа должна содержать элементы управления с пульта. Одна из кнопок пульта должна запускать автономный режим работы робота из первой части лабораторной работы.
- Сделайте вывод о работе робота в автономном режиме и режиме управления с пульта оператором.

3. Контрольные вопросы

- Расскажите про принцип действия ультразвукового датчика.

2. Какие типы датчиков также уместно применять для прохождения препятствий? Приведите примеры.
3. Изобразите алгоритм работы робота в виде блок-схем:
 - при установке одного датчика;
 - при установке двух датчиков.
4. Назовите преимущества и недостатки работы МР:
 - в автономном режиме;
 - в режиме управления с пульта.

4. Приложение. Работа со средой программирования RobotC для VexCortex

4.1. Начало работы

1. Откройте программу двойным нажатием на иконке (рис. 10.9):



Рис. 10.9. Иконка приложения

2. Перед Вами откроется окно приложения (рис. 10.10).

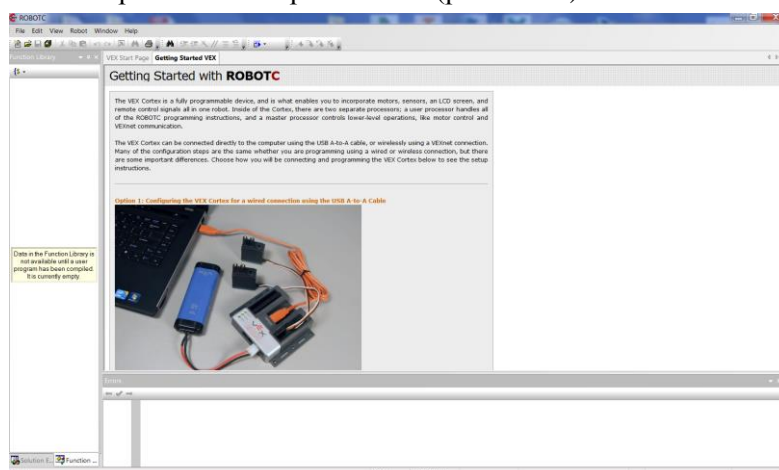
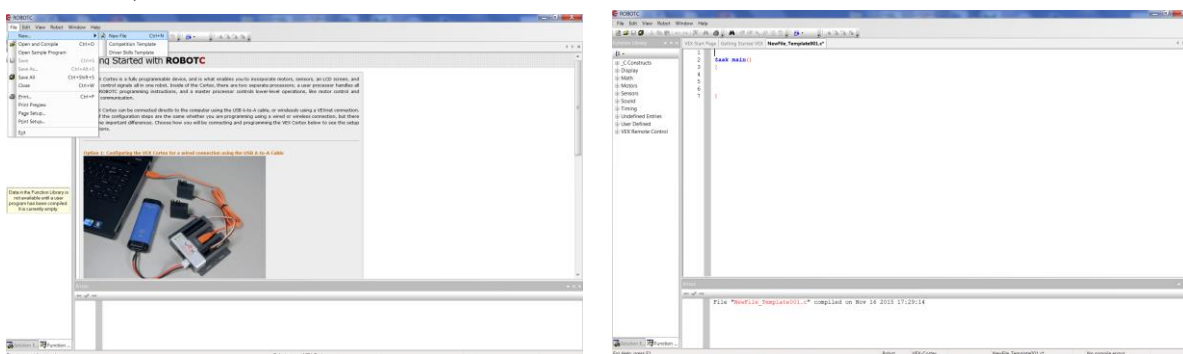


Рис. 10.10. Начальная страница приложения

3. Зайдите в контекстное меню «File» и выберите пункт «New... -> New file» (рис. 10.11, а).



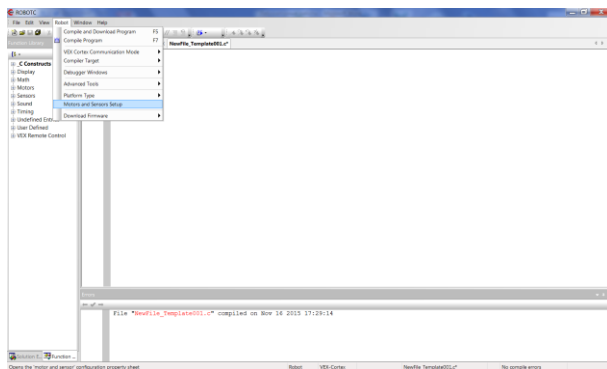
а б
Рис. 10.11. Создание нового файла (а), рабочее окно программы (б)

4. Открывается новое окно, в котором Вы будете работать над программой (рис. 10.11, б).

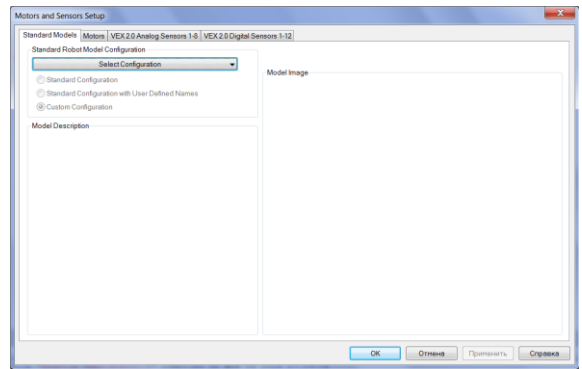
4.2. Настройка подключаемых устройств

В среде программирования RobotC для Vex Cortex придуман простой способ настройки устройств, подключаемых к контроллеру Vex.

1. Для настройки устройств зайдите в контекстное меню программы "Robot -> Motor and Sensors Setup" (рис. 10.12, б).



а



б

Рис. 10.12. Работа с подключаемыми устройствами (а), Motor and Sensors Setup (б)

2. В появившемся окне (рис. 10.12, а) есть несколько вкладок. Для настройки двигателей используйте вкладку Motors, для настройки цифровых и аналоговых датчиков VEX 2.0 Analog Sensors 1-8 и VEX 2.0 Digital Sensors 1-12 соответственно.
3. Для настройки приводов перейдите во вкладку Motors. В данном окне (рис. 10.13) вы увидите 10 строк, каждая из которых обозначает порты контроллера. В каждой строке доступны такие параметры, как:

Port – номер порта, к которому подключается привод;

Name – имя порта (присваивается для удобства работы с программой);

Type – тип привода (обычный или шаговый);

Reversed – выбор: обычное движение привода или инвертированное (при установленном флаге);

Encoder port – порт, к которому подключен энкодер, отвечающий за данный привод;

Pid Control - ПИД регулятор.

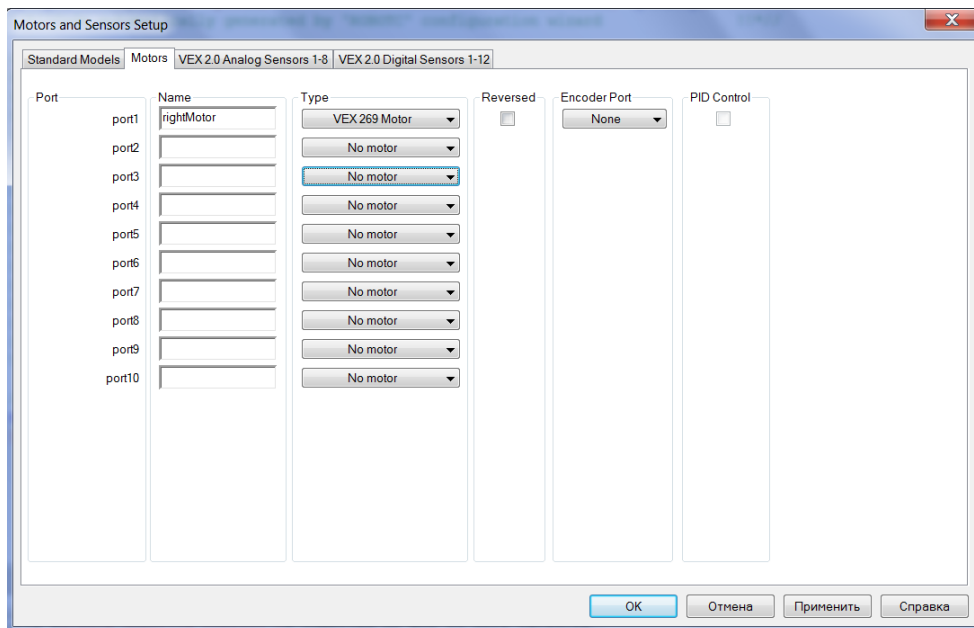


Рис. 10.13. Вкладка Motors

4.3. Первая программа в среде RobotC

Ниже приведен пример программы, написанной на языке RobotC.

```
//Описание подключенных устройств
```

```
#pragma config(Motor, port2, rightMotor, tmotorNormal, openLoop, reversed)
```

```
#pragma config(Motor, port3, leftMotor, tmotorNormal, openLoop)
```

```
task main()
```

```
{
```

```
// Робот ждет 2000 мс перед началом основной программы
```



```

        wait1Msec(2000);

// Движение вперед на полной скорости в течение трех секунд
//Двигатели подключены к портам 2 и 3
    motor[rightMotor] = 127;
    motor[leftMotor] = 127;
    wait1Msec(3000);
}
//Окончание программы и остановка робота
4) Программа для работы с ультразвуковым датчиком.
#pragma config(Sensor, dgtl8, sonarSensor, sensorSONAR_cm)
#pragma config(Motor, port2, rightMotor, tmotorNormal, openLoop, reversed)
#pragma config(Motor, port3, leftMotor, tmotorNormal, openLoop)

task main()
{
// Робот ждет 2000 мс перед началом основной программы
    wait1Msec(2000);
    while(SensorValue(sonarSensor) > 20 || SensorValue(sonarSensor) == -1)
// Цикл работает до тех пор, пока робот находится от препятствия не ближе чем 20 дюймов или до
тех пор пока в зоне видимости ничего нет (значение -1)
    {
        motor[rightMotor] = 63;
        motor[leftMotor] = 63;
    }
}

```

4.4. Программа для работы с пультом

```

#pragma config(Motor, port1, leftMotor, tmotorVex393, openLoop, reversed)
#pragma config(Motor, port10, rightMotor, tmotorVex393, openLoop)

task main ()
{
    while(1 == 1)
    {
//В зависимости от силы отклонения джойстика от начального положения, робот будет ехать мед-
леннее или быстрее
//В данной программе управление роботом осуществляется на каналах 1 и 2
        motor[leftMotor] = (vexRT[Ch2] + vexRT[Ch1])/2; // (y + x)/2
        motor[rightMotor] = (vexRT[Ch2] - vexRT[Ch1])/2; // (y - x)/2
    }
}

```