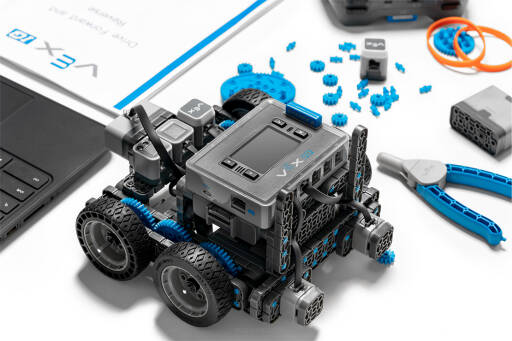
|  |
| --- |
| Робот «VEX IQ» |



|  |
| --- |
| **Оборудование:** |

Набор VEX IQ.

|  |
| --- |
| **Программное обеспечение:** |

Среда программирования ROBOTC for VEX.

|  |
| --- |
| **Реализация** |

** **

**Рисунок 1. Общий вид базовой образовательной платформы**

Для решения заданий, используйте базовую образовательную платформу, изображенную на рисунке 1. При работе с платформой проверьте подключение моторов и датчиков, к каким портам они подключены. При настройке конфигурации робота в среде программирования ROBOTC for VEX учитывайте подключение моторов и датчиков. Также проверьте установленный режим языка программирования. Для этого нужно запустить программу VEXos Utility. Робот должен быть включен и подключен к компьютеру USB-microUSB шнуром.

Решения примеров представлены рисунок 2.

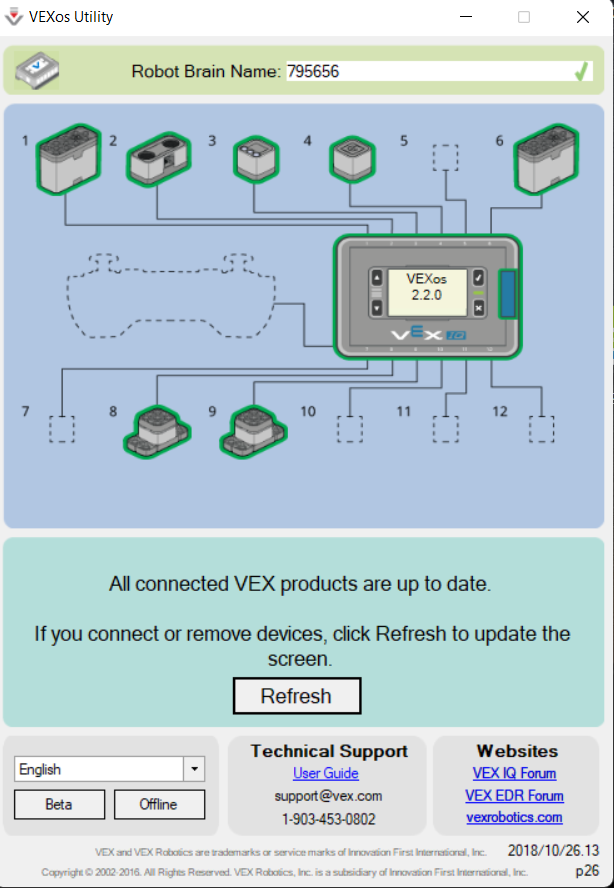


Рисунок 2

**Соблюдайте правила техники безопасности!**

Далее нужно сконфигурировать робота внутри программы ROBOTC for VEX Robotics 4.X. Для этого нужно создать New File и в вкладке – “Motor and Sensor Setup”, необходимо выбрать порты, к которым подключены моторы. И port6 поставить на реверс (рисунок 3).

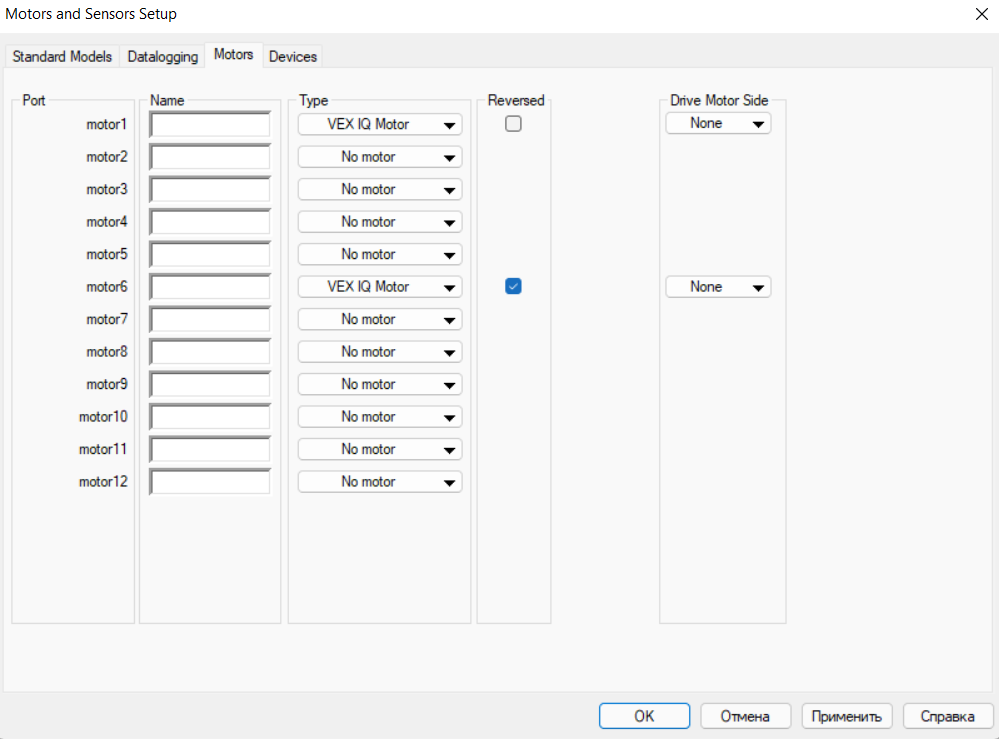
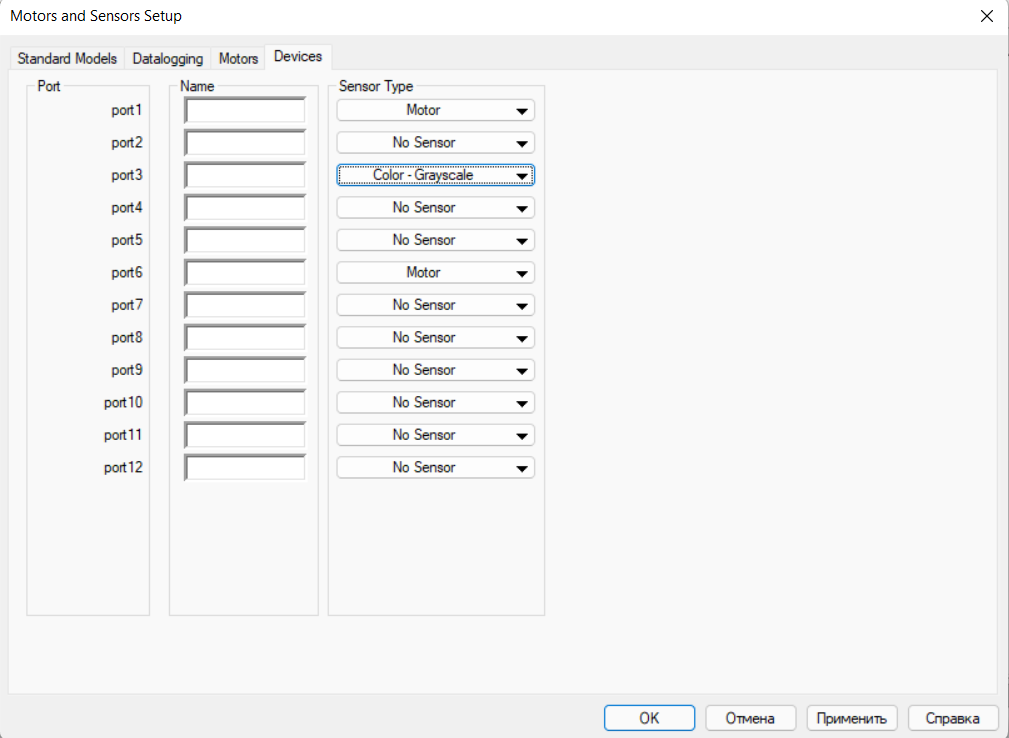


Рисунок 3

Затем, в вкладке Devices, необходимо указать датчики на определенных портах, которые будут использоваться (рисунок 4).



**Рисунок 4**

|  |
| --- |
| **Задание** |

1. Вариант 1. В среде программирования ROBOTC for VEX напишите программу для движения робота вперед в течении 2с.

#pragma config(Sensor, port3, , sensorVexIQ\_ColorGrayscale)

#pragma config(Motor, motor1, , tmotorVexIQ, PIDControl, encoder)

#pragma config(Motor, motor6, , tmotorVexIQ, PIDControl, reversed, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

task main()

{

int ML=50;

int MR=50;

motor[motor1]=ML;

motor[motor6]=MR;

wait1Msec(2000);

}

Вариант 2.

В среде программирования ROBOTC for VEX напишите программу для движения робота вперед на 1 клетку.

*Возможное решение (движение по времени, время движения подбирается эмпирическим способом):*

// настройка конфигурации робота (левый мотор подключен в порт 1, правый - в порт 6)

#pragma config(Motor, motor1, leftMotor, tmotorVexIQ, openLoop, encoder)

#pragma config(Motor, motor6, rightMotor, tmotorVexIQ, openLoop, reversed, encoder)

task main()

{

setMotorSpeed(leftMotor, 50); //левый мотор вращается с мощность 50% setMotorSpeed(rightMotor, 50); // правый мотор вращается с мощность 50% sleep(2000); //задержка 2 с

}

*Возможное решение (движение по энкодеру):*

// настройка конфигурации робота (левый мотор подключен в порт 1, правый - в порт 6)

#pragma config(Motor, motor1, leftMotor, tmotorVexIQ, openLoop, encoder)

#pragma config(Motor, motor6, rightMotor, tmotorVexIQ, openLoop, reversed, encoder)

task main()

{

//Сброс значений энкодеров.

resetMotorEncoder(leftMotor);

resetMotorEncoder(rightMotor);

//Установить вращение двигателей на мощности 75% в течение 5000 отсчетов. Количество отсчётов на единицу длины пройденного расстояния определить эмпирическим способом.

moveMotorTarget(leftMotor, 5000, 75);

moveMotorTarget(rightMotor, 5000, 75);

//Ожидать выполнения заданного перемещения.

waitUntilMotorStop(leftMotor);

waitUntilMotorStop(rightMotor);

}

2. Используя пример Detect Wall, который можно найти в среде разработки ROBOTC for VEX, напишите программу для движения робота вперед и его остановки перед препятствием на расстоянии 15 см.

3. Напишите программу для поворота робота на 900 по часовой стрелке. Для точного поворота робота на заданный угол можно использовать энкодеры двигателей и/или гироскоп.

Чек-лист

|  |  |  |  |
| --- | --- | --- | --- |
| Номер задания | Задание | Критерии | Результат |
| 1 | Включение и проверка конфигурации робота. | 0 – не выполнено  1 - выполнено |  |
| 2 | Написание и проверка кода. | 0 – не выполнено  1 - выполнено |  |
| 3 | Загрузка кода в робота. | 0 – не выполнено  1 - выполнено |  |
| 4 | Проезд робота по прямой в течении  2 секунд. | 0 – не выполнено  1 - выполнено |  |
| 5 | Расчет числа оборотов колеса по формуле: L=2ПR | 0 – не выполнено  1 - выполнено |  |