

СЦЕНАРИЙ УРОКА ДВОИЧНОЕ КОДИРОВАНИЕ. SWITCH CASE

Цель урока: познакомиться с принципами двоичного кодирования и программной конструкцией switch case.

Результаты:

- знакомство с понятием двоичного кодирования;
- умение эффективно использовать вложенные ветвления if else и программную конструкцию switch case;
- умение организовать работу с пультом дистанционного управления;
- формулирование выводов по результатам эксперимента.

Формируемые компетенции:

предметные:

- умение подключить микроконтроллер VEX IQ к компьютеру;
- умение использовать конструкцию switch case;
- умение подключить пульт дистанционного управления;
- умение применять двоичное кодирование;
- умение запустить программу;
- овладение методами моделирования, конструирования и эстетического оформления изделия;
- умение работать по инструкции;

метапредметные:

- умение устанавливать взаимосвязь знаний по разным учебным предметам для решения прикладных учебных задач;
- умение соблюдать условия эксперимента для получения наиболее точных результатов;
- умение ориентироваться на заданные критерии;
- умение выбрать из нескольких решений более эффективное;
- работа с информацией и использование ресурсов;
- умение проводить оценку и испытание полученного продукта;
- умение формулировать выводы по результатам эксперимента;

личностные:

- готовность и способность вести диалог и достигать в нем взаимопонимания;
- освоенность социальных норм, правил поведения, ролей и форм социальной жизни в группе;
- способность к совместной работе ради достижения цели;
- умение анализировать, проектировать и организовывать деятельность;
- способность принимать решения.

Необходимые материалы:

- конструктор Vex IQ (из расчета одна команда - один набор, две команды - два набора и т.д.);
- индивидуальный рабочий лист, распечатанный для каждого ученика;
- персональные компьютеры (по количеству учащихся);
- компьютер и проектор для демонстрации справочного видео.

Ход урока:

Обсуждение темы урока:

1. Предыдущий урок был посвящен вопросу декомпозиции в механике и сравнению разных видов привода. На данном уроке будет разобрано двоичное кодирование и проблема реализации большого количества вложенных ветвлений через программную конструкцию **switch case**.
2. **Загадайте** число от 0 до 32 и предложите ребятам угадать. Часть ребят начнут называть конкретные числа. Предложите им попробовать отсечь какое-то количество чисел одним вопросом.
3. Если ребятам будет сложно догадаться самостоятельно, попросите класс загадать число. И задавайте им вопросы, считая их количество:
 - 1) Число меньше 16? Предположим, ответ да.
 - 2) Число меньше 8? Предположим, ответ да.
 - 3) Число меньше 4? Предположим, ответ нет.
 - 4) Число меньше 6? Предположим, ответ нет.
 - 5) Число меньше 7? Предположим, ответ да. Это число 6.
6. Может быть, вопросов получится меньше, но самое большое количество вопросов для 32 чисел - 5. Напомните, что 1 вопрос = 1 датчик = да/нет = 1/0 и такое количество информации называется 1 бит.
7. Считается количество вопросов (датчиков) по формуле:

$$\begin{array}{l} \text{Количество датчиков} \\ \text{Количество вопросов датчиков} \\ \times \\ \text{Количество датчиков} \\ \text{Количество вопросов датчиков} \\ \hline \text{Количество состояний} \end{array}$$

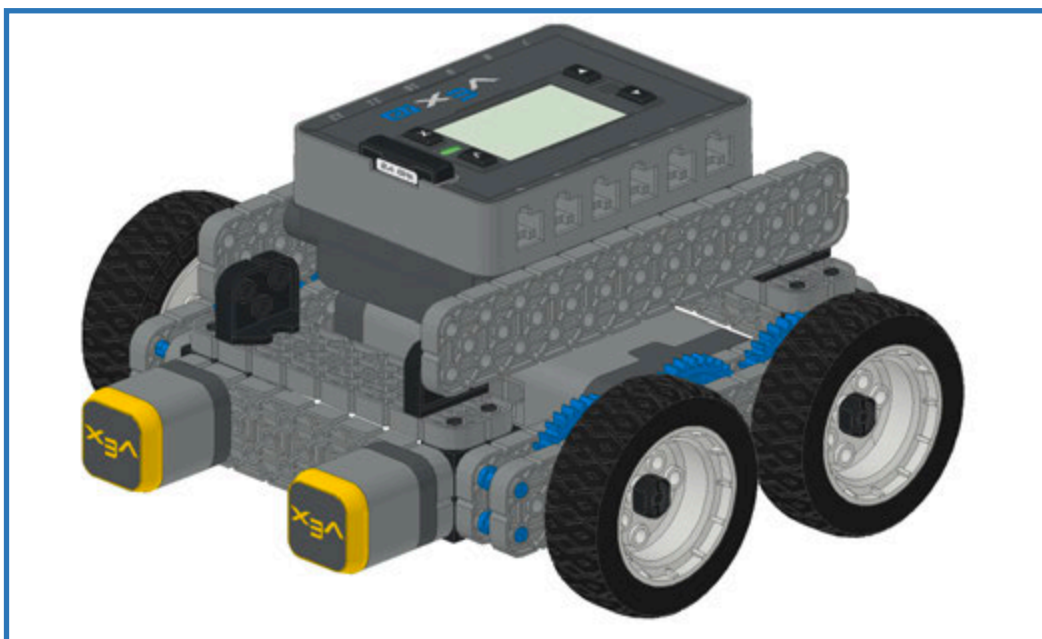
В математическом виде эта формула выглядит так: $2^B \geq C$ в степени количества вопросов (**B**) должно быть больше количества состояний (**C**):

$$2^B \geq C$$

8. **Предложите** учащимся посчитать по этой формуле, сколько датчиков-кнопок понадобится, чтобы запрограммировать все состояния одного двигателя: вращение вперед, вращение назад и остановка.
9. **Попросите** записать результаты в рабочих листах (задание 1.2).
10. Даже для программирования такой простой «угадайки» понадобилось бы целых пять ветвлений, вложенных друг в друга. Такая программа уже очень большая и сложная. А если состояний, например, 256? На помощь приходит программная конструкция switch case.
11. **Предложите** учащимся запрограммировать автоматические поворотники для тележки с помощью конструкции switch case. Если робот начнет останавливаться, оба датчика должны сменить цвет на красный; при повороте направо желтым цветом должен загореться правый датчик, налево – левый. При движении вперед датчики должны быть погашены.
12. **Попросите** учащихся записать цель занятия (так, как они ее поняли) в рабочих листах (задание 1.1).

Этап конструирования и программирования:

13. **Предложите** учащимся разделиться на команды и собрать тележку с двумя датчиками касания с LED подсветкой по инструкции или самостоятельно.



14. Для того чтобы приступить к программированию, **попросите** учащихся распределиться за компьютерами и открыть RobotC.

15. **Создайте** новый файл в среде программирования RobotC.
16. **Каждой команде необходимо** запрограммировать работу тележки от пульта дистанционного управления. Для этого можно использовать программу с предыдущего урока (более того, это приветствуется). Однако необходимо внести следующие изменения:
 - создать функцию `int pult2b()`, внутри которой поместить уже написанные ветвления. Эта функция будет возвращать номер состояния, в котором находится робот. Всего этих состояний 4: робот едет вперед, поворачивает налево, поворачивает направо, стоит на месте;
 - добавить первое состояние «робот едет вперед» в 17 строку - `return (1)`;
 - добавить второе состояние «робот поворачивает направо» в 22 строку - `return (2)`;
 - добавить третье состояние «робот поворачивает налево» в 30 строку - `return (3)`;
 - добавить четвертое состояние «робот стоит на месте» в 35 строку - `return (4)`;
 - в `task main()` должен остаться бесконечный цикл `while (1)`, а в теле цикла функция вывода на экран `displayVariableValues (1 , pult2b())`.

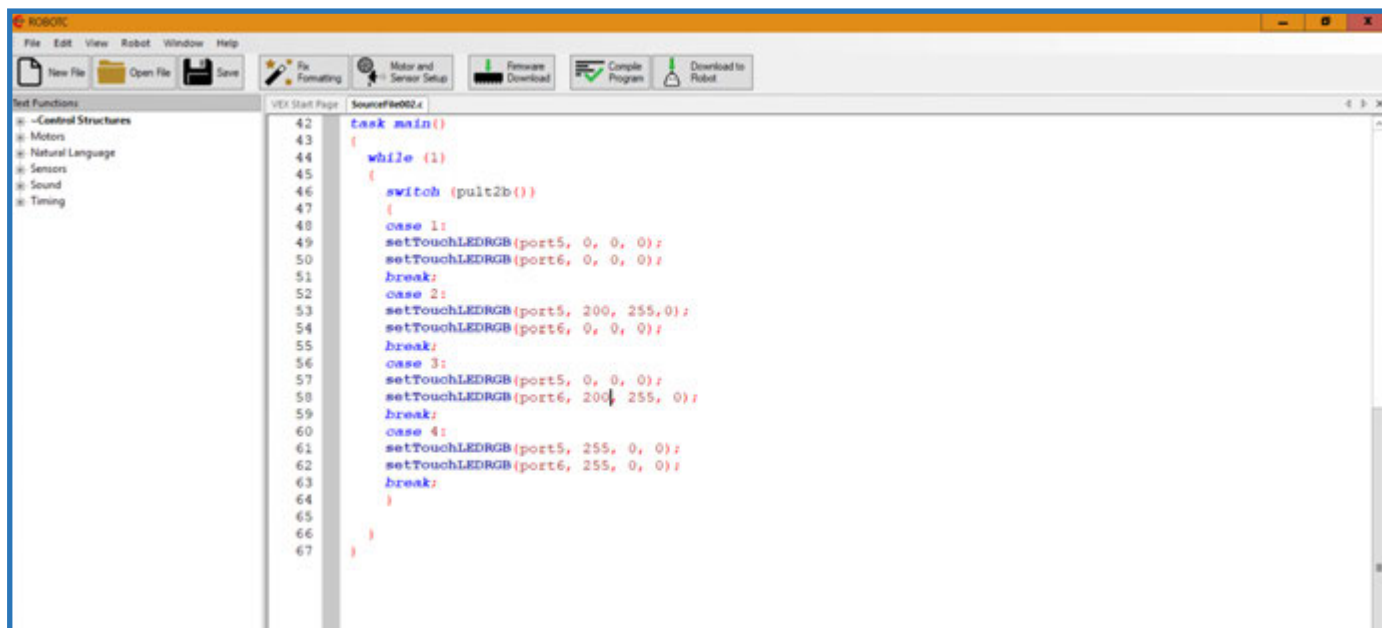
```

10 int pult2b ()
11 {
12     if (getJoystickValue (BtnLUp))
13     {
14         if (getJoystickValue (BtnRDp))
15         {
16             move (100,100,1);
17             return (1);
18         }
19         else
20         {
21             move (100,0,1);
22             return (2);
23         }
24     }
25     else
26     {
27         if (getJoystickValue (BtnRDp))
28         {
29             move (0,100,1);
30             return (3);
31         }
32         else
33         {
34             move (0,0,1);
35             return (4);
36         }
37     }
38 }
39 task main()
40 {
41     while (1)
42     {
43         displayVariableValues(1,pult2b());
44     }
45 }
  
```

17. **Подключите робота к компьютеру и сделайте инициализацию датчиков и моторов с помощью утилиты VEX OS Utility.**
18. **Каждая команда должна** проверить готовность своего робота к тестированию. Сделать это можно, выполнив задание 2.1 в рабочих листах.
19. **Предложите** проверить работоспособность программы.
20. Осталось добавить LED-подсветку. Для этого необходимо стереть функцию `displayVariableValues (1 , pult2b())` и написать в тело цикла следующий код:
21. `switch(pult2b())` - вызов функции `switch case`, а также функции `pult2b()` управления тележкой. В теле функции `switch` описываются действия, которые необходимо совершить при получении значения 1, 2, 3 или 4 от функции `pult2b()`. Например, для движения вперед в 48 строке указываем

значение 1, в этом случае светодиоды необходимо выключить (строки 49-50); для поворота направо в 52 строке указываем значение 2, в этом случае необходимо левые светодиоды выключить (54 строка), а правые включить так, чтобы они показывали желтый цвет (53 строка), и т.д. Каждый вариант отделяется друг от друга фразой **case *значение***; и заканчивается командой **break**.

- * Обязательно в ходе урока задайте каждому ученику несколько вопросов о назначении команд, о цели его программы, чтобы убедиться в достаточном уровне понимания материала.
- * Если у команд возникнут трудности с составлением программы, можно пример программы вывести на экран.



```
task main()
{
  while (1)
  {
    switch (pult2b())
    {
      case 1:
        setTouchLEDRGB(port5, 0, 0, 0);
        setTouchLEDRGB(port6, 0, 0, 0);
        break;
      case 2:
        setTouchLEDRGB(port5, 200, 255, 0);
        setTouchLEDRGB(port6, 0, 0, 0);
        break;
      case 3:
        setTouchLEDRGB(port5, 0, 0, 0);
        setTouchLEDRGB(port6, 200, 255, 0);
        break;
      case 4:
        setTouchLEDRGB(port5, 255, 0, 0);
        setTouchLEDRGB(port6, 255, 0, 0);
        break;
    }
  }
}
```

Этап проведения эксперимента:

22. **Предложите** командам протестировать тележку на предмет точного соответствия программы условиям задачи, то есть убедиться, что все кнопки пульта, а также LED-подсветка работают в соответствии с задумкой и тележка может ехать вперед, разворачиваться влево и вправо, останавливаться.
23. **Попросите** занести результаты тестирования в рабочих листах (задание 2.2).

Этап рефлексии:

24. **Обсудите с учащимися**, как они понимают, для чего нужно двоичное кодирование? Когда использовать **switch case**, а когда вложенное ветвление? Почему?
25. **Предложите** учащимся, ориентируясь на сформулированную в начале урока цель, записать в рабочих листах (задание 3.1), чему они научились сегодня.

Этап приведения кабинета в порядок:

26. Предложите ребятам разобрать тележки следующим образом: разобрать на своем рабочем месте все детали и разложить их по видам; каждый вид положить в отдельную ячейку в коробке с конструктором.