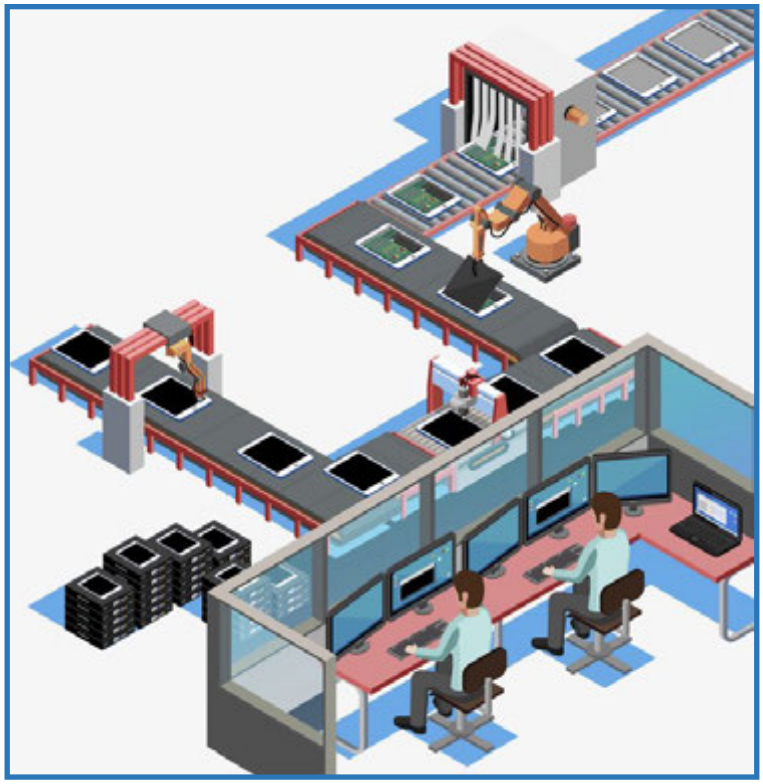
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ
ЯЗЫК ПРОГРАММИРОВАНИЯ РОБОТОВ RobotC

Важнейшим достижением человечества, одной из ключевых технологий, которые изменили нашу жизнь, стала стандартизация. Действительно, чрезвычайно важно уметь создавать такие производственные особенности, которые можно многократно повторять. Как мы уже неоднократно говорили, сутью технологии и является описание такой последовательности, в результате воздействия которой на ресурсы всегда получаются продукты с одинаковыми свойствами. Изначально эту задачу решали, разбивая технологический процесс на подзадачи, с которыми справлялись участники производственного процесса. Так появился конвейер – способ производства, где каждый человек выполняет свою маленькую задачу, естественно, что в этой задаче человек становится истинным мастером.



Но вскоре выяснилось, что часть простых задач лучше и надежнее с точки зрения повторяемости выполняют машины. Например, если необходимо завернуть гайку на строго определенный угол. В дальнейшем машины совершенствовались, приобретая все больше самостоятельности. Позже люди освоили новую технологию – робототехнику, но

об этом мы поговорим позже. Сейчас же в нашу задачу входит познакомиться с новой технологий, ориентированной на автономность машин, - программированием.

Что это за технология - программирование?

Программирование – это создание инструкций, основанных на алгоритмах обработки данных, позволяющих вычислительным устройствам выполнять вычисления или функции управления.

Это большое определение требует пояснения. Прежде всего, нам встретился новый термин - алгоритм.

Алгоритм - это набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.

Обратите внимание на то, что термины «алгоритм» и «технология», по сути, являются родственными, но «алгоритм» используется в более строгих рамках - в случае, если можно говорить о строго гарантированном результате.

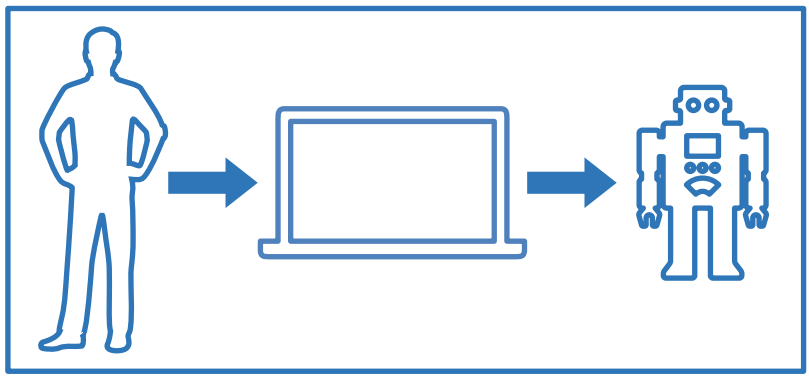
В программировании алгоритмы совершают операции с данными. В общем и целом, данные в программировании - это, как правило, числа. Понятно, что инструкция позволяет производить преобразования с этими числами или принимать управляющие воздействия в зависимости от этих преобразований.

Для того чтобы реализовать технологии программирования, разработаны *языки программирования*. В языках программирования зарезервировано определенное количество функций в виде специальных слов и символов. Набор таких слов и символов, записанных по правилам языка программирования, называется *кодом программы*. Выражения «писать код» и «программировать» являются тождественными.

Все языки можно разделить на две группы:

1) языки, которые осуществляют перекодирование программы в машинный код для ее выполнения, называются *компиляторы*;
2) языки, выполняющие инструкции программы по-командно, не перекодируя ее, называются *интерпретаторы*.

Для написания наших программ мы будем использовать разновидность одного из самых распространенных языков программирования - C++. Этот язык программирования является компилятором. То есть сначала мы будем записывать код программы, затем он будет преобразовываться (компилироваться) в код, понятный процессору нашего робота, затем мы будем загружать код в робота и запускать его в нем.



Наш язык программирования называется RobotC. Этот язык широко распространен среди робототехников по нескольким причинам. Во-первых, он

охватывает очень большое количество робототехнических платформ: Vex, Lego, Arduino. Во-вторых, он является бесплатным. В-третьих, в случае, если у вас нет робота, RobotC придет к вам на помощь: в нем есть виртуальная среда, где можно опробовать поведение робота.

Язык программирования С основан на применении функций.

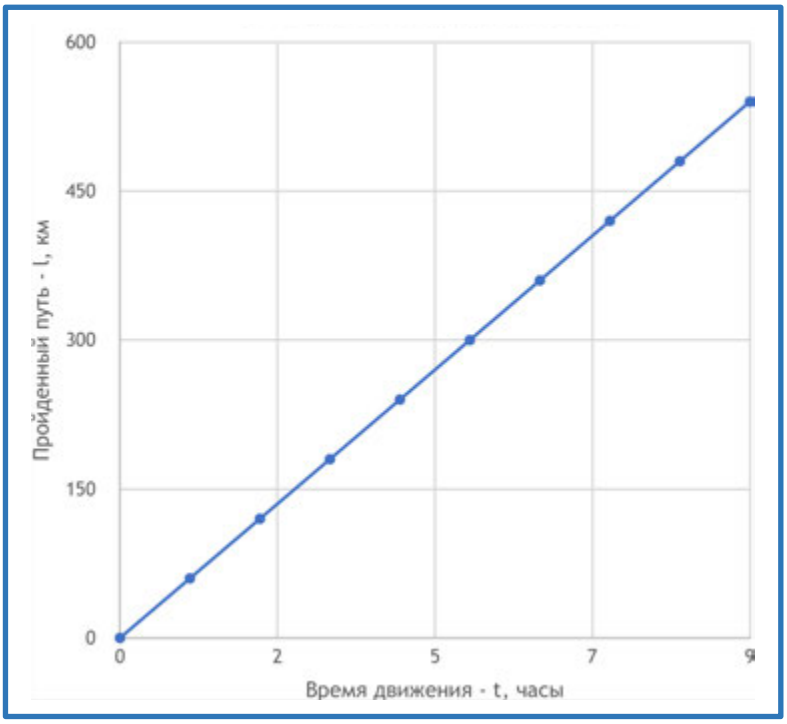
Функция - это соответствие между элементами двух множеств, установленное по такому правилу, что каждому элементу одного множества ставится в соответствие некоторый элемент из другого множества.

Функции очень распространены в математике и являются удобным инструментом для решения множества задач. Например, автомобиль едет строго по прямой с постоянной скоростью. Как описать положение автомобиля в любой момент времени? Мы знаем, что пройденный путь связан с временем движения в этом случае следующим образом:

Эти же данные можно представить в таблице, например, для скорости автомобиля в 60 км/ч:

Время t, часы	0	1	2	3	4	5	6	7	8	9
Пройденный путь *l, км*	0	60	120	180	240	320	360	400	480	540

Достаточно наглядно, но не слишком удобно. Что, если необходимо узнать, где оказался автомобиль через 3,5 часа движения? В этом случае можно путь, пройденный автомобилем, представить и в виде графика:



Воспользовавшись линейкой, на этом графике можно найти пройденный путь для любого отрезка времени.

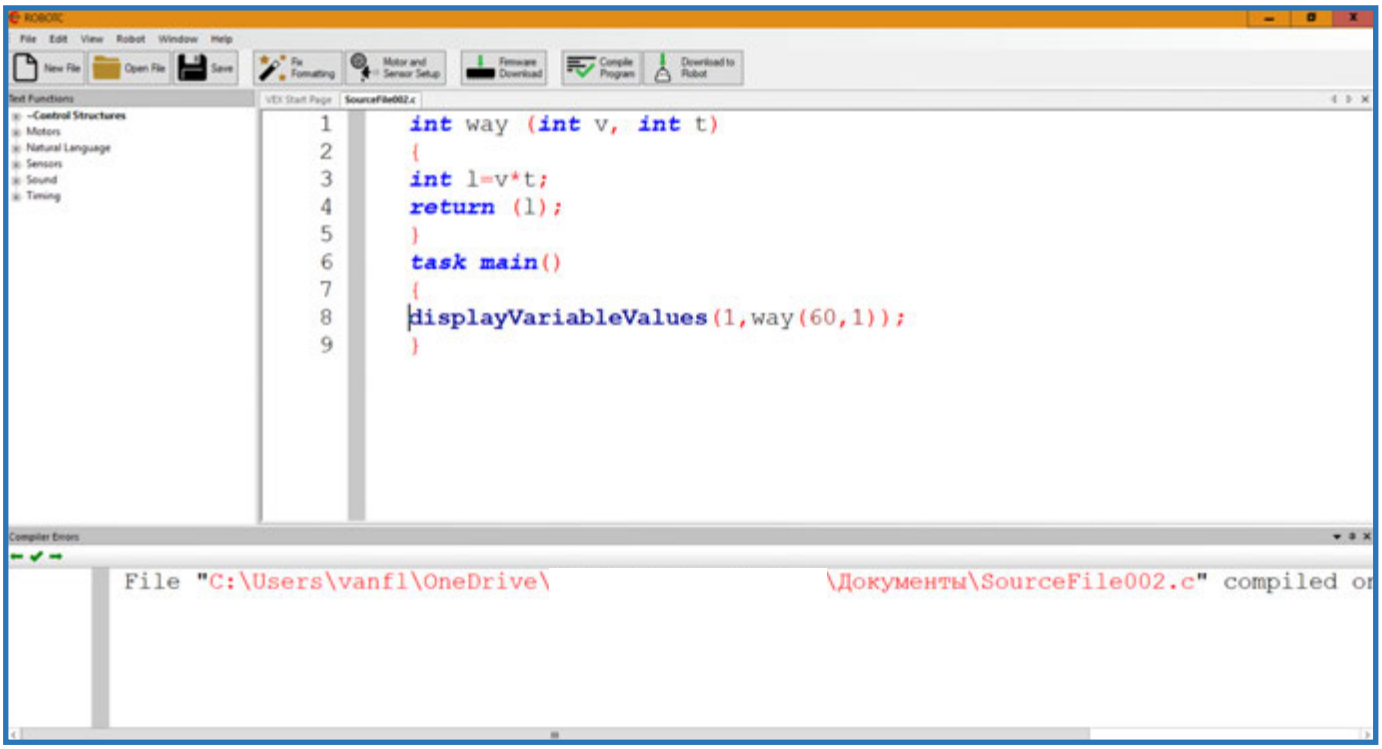
Обратите внимание на то, что в данном примере присутствует два множества: множество промежутков времени от начала движения, и множество пройденных путей. Оба эти множества связаны между собой при помощи величины скорости. У нас получилась самая простая функция, которая называется линейной (ведь график этой функции – линия). Эту функцию записывают следующим образом:

$$l(t) = v * t$$

В этой записи видно, что путь l зависит только от аргумента функции t.

Важно! Только что мы с вами познакомились с новой технологией – технологией представления информации: мы рассмотрели алгебраическое, табличное, графическое и функциональное представление.

В языке С используется функциональная форма представления. На языке, понятном компьютеру, это выглядит так:



В первой строке программы объявляется, что нами будет создана функция с именем **way**. **int** означает, что данная функция после выполнения должна будет возвратить целое число, то есть **int** – это тип функции. Значения в скобках – это аргументы функции и их типы. В нашем случае используется два аргумента типа **int** (целое число).

После того как были записаны название и тип функции, а также названия и типы аргументов, необходимо описать тело функции.

Тело функции – это описанные в порядке очереди действия с аргументами функции.

Описание тела функции находится между фигурными скобками, открывающимися во второй строке и закрывающимися в пятой. Между этими строками есть четыре команды. В третьей строке располагаются сразу две команды. Первая - это объявление переменной `l` (`int l` означает, что эта переменная обязательно будет целым числом).

Переменная - это именованная сущность, которая может изменять свое значение.

Обратите внимание на то, что в нашей работе пока используются только целочисленные переменные.

Вторая команда в третьей строке – это команда присваивания. Переменной `l` присваивается значение (присваивание в языке программирования С – обозначается символом =).

Третья операция – это операция умножения аргументов v и t (арифметические операции в С:

+ сложение
- разность
* умножение
/ деление).

В третей же строке после всех команд находится символ «`;`». Точка с запятой показывает, что заканчивается один набор команд и начинается следующий.

В четвертой строке находится команда `return (l)`. Обратите внимание: она оформлена в точности как функция, но у нее нет аргумента! Это связано с тем, что эта команда, по сути, является внутренней функцией языка С. Ее тело уже описано в языке, и нам необходимо только воспользоваться результатом работы этой функции. А результат – это так называемое возвращаемое значение. Значение, которое отдает функция для дальнейших преобразований.

В шестой строке новая функция. Она является основной задачей. Поэтому она и называется `main`, а ее тип - `task`. У этой функции не может быть аргументов и возвращаемых значений.

В седьмой строке находится начало этой функции, в девятой – конец. Выполнение программы, после того как она будет загружена в робота, как раз и начнется с седьмой строки.

В восьмой строке находится функция `displayVariableValues()` – это специфическая функция для языка программирования RobotC. Она выводит на экран значение переменной и имеет два аргумента `displayVariableValues(lineNumber,value)`. Первый – номер строки дисплея блока VEX IQ, второй – имя переменной. В нашем случае в роли переменной выступает целая функция. То есть получается, что функция вызывает функцию! Давайте проследим, как это происходит.

Вызывается функция `way` с аргументами `(60,1)`. Эта функция присваивает переменной v значение первого аргумента, и переменной t – значение второго аргумента.

Далее эти переменные перемножаются, и полученное значение присваивается переменной `l`. Это значение и возвращает функция `way`. Возвращенное значение функции передается в функцию `displayVariableValues(1,way (60,1))`, которая, в свою очередь, выводит в первую строку возвращенное значение.

Итак, язык С является языком функций. Для его правильного использования необходимо уметь описывать функции, понимать, как они взаимодействуют между собой, уметь вызывать их, понимать, что одна функция может вызывать другую.