

Привет ребята. В этом выпуске уроков ардуино расскажу вам про условный оператор и оператор выбора.

Условный оператор очень важная штука, без которой практически невозможно представить никакую программу. Условный оператор иф, с переводе с английского если, проверяет истинность условия, которое находится в скобках сразу после него. Если условие правдивое, или равно логической единице, то выполняется кусок кода, заключенный в идущих дальше фигурных скобках. Если условие не выполняется, или в скобках логический 0, то весь блок в фигурных скобках игнорируется.

Условие задается напрямую, или при помощи логических операторов. Напрямую это например есть у нас логическая переменная, равная 0 или 1. Ставим имя этой переменной как условие, и если переменная на данный момент равна 1, то выполняются команды в фигурных скобках. Дословно это читается так: если выражение в скобках правдиво, то выполнить команды. Или: если выражение в скобках равно единице, то выполнить команды.

Эта запись равносильна сравнению с единицей, сравнение на языке ардуино это двойной знак равно. Сравнение двух переменных возвращает нам результат в виде правда или ложь. То есть если $5 > 4$, то мы получаем правду, или логическую единицу. И наоборот, запись вида $4 > 5$ вернёт нам 0. Мой пример читается так: если переменная val равна единице, то выполнить команды.

Помимо равенства существуют и другие логические операторы, такие как не равно, больше, меньше, больше или равно и меньше или равно. Их можно использовать для обработки значения с датчиков, например если значение с датчика больше чем 300, то выполнить код. Похожие конструкции используются очень часто.

Так как условие строится по законам математической логики, оно может быть сложным. Например мы хотим совершать какое то действие при значении датчика от 200 до 400. Для понимания отмечу их на числовой прямой. Для создания такого условия нужно дать два условия: если значение больше 200, и значение меньше 400. Между условиями нужно поставить логическое и, которое на языке ардуино выглядит как двойной знак энд. Читается это так: если значение с датчика больше 200 и значение с датчика меньше 400 то выполнить код в фигурных скобках. Если же мы хотим попасть вот в эти области значений, то используем оператор или, который выглядит как две вертикальные линии. Читаем: если значение меньше 200 или больше 400 то выполнить код. Условие может быть настолько сложным, насколько вам это нужно, главное не запутаться и везде ставить скобочки.

Вернёмся к оператору иф. Можно продлить конструкцию, добавив слово елс, в переводе означает иначе. То есть при невыполнении условия будет выполнен блок кода, следующий за словом else.

Давайте получим данные с порта и обработаем их при помощи условия. Как работать с портом я рассказывал в предыдущем выпуске уроков ардуино, поэтому момент принятия значений опускаю.

Пример: если значение, принятое из порта равно единице или цифре 5, то отправляем в порт строчку. Иначе будет отправлена другая строчка. Вооот. А если нам нужно больше вариантов? Есть конструкция елс иф, которая будет проверять новое условие при невыполнении предыдущего. То есть это все читается вот так: если переменная равна 1 то отправить первую строчку, иначе если она равна 2, отправить вторую строчку, иначе если трем, то третью. А если переменная не равна 1 2 или 3, то отправить четвертую строчку. Проверяем.

Данная конструкция выглядит очень громоздко, и можно заменить ее другим очень удобным оператором выбора, свитч кейс. После слова свитч в скобках указывается переменная, которую будем проверять. И далее следует набор вариантов, обозначаемых словом кейс. Идет сам вариант, и ставится двоеточие. Далее пишем набор команд, которые нужно выполнить. И в конце обязательно ключевое слово брейк, чтобы выйти из свича. Читается это следующим образом в случае если переменная val равна 1, выполнить этот кусок кода, если равна 2 то выполнить этот, и так далее. Еще можно указать дефолт, следующий код будет выполнен если не выполняется ни одно из предыдущих условий. Можно переписать наш обработчик данных с порта, используя оператор свитч кейс.

Используя эту конструкцию, можно делать текстовое меню для вашего устройства на ардуино, как например это сделано у моего хронографа. Выбор режимов и все такое. Нужно просто много вложенных друг в друга условий, и желание не запутаться в этой паутине условий)

Сделаем вывод: при помощи условного оператора можно обрабатывать значения с датчиков, можно переключать режимы работы устройства, можно даже сделать текстовое меню и много много других классных вещей. Спасибо за внимание, с вами был канал заметки ардуинщика, до встречи в новых уроках.