

Всем привет, в этом выпуске мы продолжим знакомиться с ардуино. В прошлом выпуске рассказывал про ардуино с точки зрения железа и подключения проводов, а сегодня начнем осваивать программную часть, в том числе типы данных.

На самом деле между этим и прошлым выпуском есть еще один, он называется начало работы с ардуино и тоже есть на канале. Там я показал как скачать, установить и настроить софт для работы с ардуино.

Запускаем значит ардуино иде. Перед нами открывается стандартный пустой кусок кода. Сохраним его. Прошивки для ардуино называются скетчами. Скетчи имеют расширение Ино, и хранятся в папке с таким же названием как сам скетч. Если вы запустите прошивку без папки, то папка будет автоматически создана и файл будет в нее перемещен.

Начнем разбирать синтаксис. Кстати вот вам совет: заведите тетрадочку, и вписывайте в нее команды с пояснениями. Чтобы программировать на ардуино, нужно знать на память команды, или как минимум держать их перед глазами. Записывая в тетрадку, вы быстрее все запомните, да и работать по своему конспекту будет намного удобнее.

Ардуино программируется на языке ардуино вайринг, который является упрощенным с++ и наследует от него синтаксис. Когда вы нажимаете кнопку загрузки, прошивка компилируется и превращается в машинный код, который загружается в ардуино. Этот процесс необратим, то есть достать прошивку из прошитой ардуино в виде вот такого скетча - уже невозможно.

Структура прошивки примерно следующая: в самом начале можно оставить для себя заметку, что это за программа и какие компоненты используются. Однострочные комментарии отмечаются двойным слешем. Все написанное в строчку после двойного слеша будет проигнорировано при компиляции, то есть комментарии места в памяти не занимают. Кстати таким же образом можно убирать строки из программы при отладке. Многострочный комментарий оставляется вот так. Кстати запомните, что программа выполняется сверху вниз.

Далее подключаются библиотеки, объявляются переменные и константы, и далее идет void setup. Словом void объявляется функция, которая не возвращает никаких значений. Функция сетап является стандартной, все что находится внутри неё будет выполнено один раз при каждом запуске ардуино. Зачем это нужно? Здесь прописываются настройки и режимы работы самой ардуино, также инициализируются датчики и модули, и прописываются их настройки и режимы работы. Грубо говоря проходит подготовка к запуску основной программы.

Далее идет основной блок, цикл лууп. Все что находится внутри него, будет выполняться сверху вниз, в бесконечном цикле. Именно здесь происходит основное действие, считываются значения с датчиков, кнопок, крутилок, идут расчеты и дальнейший вывод на дисплей, управление лампочками моторчиками и все такое.

Также в этот выпуск хочу включить типы данных. Это скучная тема, я сначала не вникал в нее всерьез, но потом пришлось, так как незнание типов данных ведет к ошибкам, которые очень сложно потом найти и исправить.

Зачем же нужны переменные разных типов? Во первых, все типы занимают разный объем памяти, а нам каждый байт важен. И во вторых, имеют разное предназначение. То есть мы максимально точно

сообщаем микроконтроллеру, чего мы от него хотим, и он будет выполнять действия быстрее при правильном использовании разных типов данных.

Переменная объявляется следующим образом: сначала пишется тип данных. Далее идет название переменной, и в конце строки ставится точка запятой, как и в конце каждой команды. По умолчанию переменная имеет значение 0, но можно присвоить ей любое другое значение при помощи знака равно. Если вы присвоите переменной значение, которое не входит в диапазон типа данных, то получите ошибку или ваша программа будет работать некорректно.

Начнем с того, что вы должны помнить со школы. Информация измеряется в битах, где бит это элементарная ячейка памяти. Самый маленький тип данных для ардуино - булин, занимает 8 бит, или 1 байт, и может быть равен 1 или 0. Особенность Булина в том, что он является логическим типом, то есть может принимать значения правда или ложь, это то же самое что 1 и 0. И самое удобное, что его значение очень удобно инвертировать, при помощи восклицательного знака. Переменные типа булин обычно используется для хранения флажков и состояний, об этом будет отдельное видео, так как вопрос очень важный.

Следующий тип - чар, по объему тоже занимает 1 байт, и может принимать значение от -128 до 127. Используем для хранения одного текстового символа, при помощи одинарных кавычек, причем символ хранится в виде числа согласно таблице символов.

Следующий тип тоже занимает 1 байт и так и называется, байт. Хранит значение от 0 до 255, используется для хранения небольших чисел.

Следующий тип самый популярный и используется чаще всех, инт. Занимает 2 байта и хранит в себе число от - до + 32000 с хвостиком тысяч. Еще есть вариант unsigned инт, он хранит в себе только положительные числа, до 65000.

Следующим тип данных - ворд, абсолютно идентичен предыдущему.

Для хранения совсем больших чисел есть тип Лонг, занимает он 4 байта и может хранить в себе число чуть больше двух миллиардов в обе стороны от нуля, соответственно ансайд Лонг хранит положительное число до 4 миллиардов.

Тип флоат используется для хранения чисел с плавающей точкой, то есть десятичных дробей, и занимает те же 4 байта. Тип дабл точнее чем флоат, но на ардуино они имеют одинаковую точность, так что работаем только с флоат, и дабл не используем. Здесь тоже есть свои особенности, например переменной с типом флоат присваивать нужно обязательно число с точкой, даже если оно целое.

Если вы хотите разделить число типа флоат на другое число, то делить нужно опять же на число с точкой, иначе результат будет целым числом. А в том случае если мы делим переменную целого типа и хотим получить дробное значение, то перед ней нужно поставить слово флоат в скобках. Это нужно для того чтобы компилятор выделил из памяти 4 байта для проведения вычислений десятичных чисел. Кстати операции с числами типа флоат занимают намного больше времени чем с остальными типам данных, так что использовать надо с умом. Иногда выгоднее умножить числа на тысячу, посчитать все в больших числах, а потом снова разделить на тысячу. Отметьте это себе. Также можно выводить числа с указанным количеством знаков после запятой, но этим мы займемся

в другом выпуске. Еще есть массивы и строки, но про них тоже стоит поговорить отдельно. Так что на сегодня все! В следующем выпуске научимся работать с ком портом для общения с ардуино.