

Всем привет, продолжаем цикл уроков об ардуино и сегодня расскажу вам про встроенный канал общения с ардуинкой, сом порт, он же серийный порт, он же монитор порта, он же консоль.

Монитор порта открывается вот этой кнопкой в программе ардуино иде, с его помощью мы можем посылать на ардуинку всякие значения, а также ардуинка может слать чего нибудь нам. Это могут быть как цифры, так и текст. Но чтобы это произошло, ардуинку естественно нужно запрограммировать.

Для работы с портом есть стандартная библиотека, которая имеет объект сериал, пишется с большой буквы. Видите, цвет изменился. Чтобы работать с портом, сначала нужно его открыть. Делается это в блоке сетап, чтобы порт открылся при запуске ардуино, при помощи команды сериал.бегин, в скобках указывается скорость связи с ардуинкой. Измеряется она в бодах, что это такое вы можете почитать в гугле. Стандартный набор скоростей можно посмотреть в настройках в самом мониторе порта. По умолчанию тут стоит скорость 9600 бод, поэтому её и будем использовать.

Чтобы отправить что то в порт, используется метод принт, как и все методы он пишется через точку после объекта. Давайте выведем хелло ворлд. Чтобы вывести именно текст, заключаем его в двойные кавычки. Мы написали эту команду в блоке сетап, так что строчка будет выведена 1 раз при запуске ардуино. Прошиваем. Кстати при открытии порта ардуино перезагружается. Отлично. Выведем еще чего нибудь. Смотрите, кириллицу монитор не любит. Для того, чтобы выводить кириллицу, нужно скачать сторонний монитор порта, например путти. И второе наблюдение, строчки склеились. Чтобы автоматически перевести строку после вывода, есть команда принтлн. Вот вам еще пример использования, хочу вывести две строчки, и каждая состоит из нескольких частей. Заметьте, что цифры я отправляю без кавычек. Цифры и переменные отправляются без кавычек. О, давайте еще выведем строку, то есть строку как тип данных. Объявим строку и присвоим ей пару слов.

Еще одна полезная штука вывода в порт, возможность указать число знаков после запятой. Объявим переменную типа флоат, с кучей знаков после запятой. Число знаков при выводе указывается через запятую после имени переменной. Давайте выведем только три знака. Ещё после запятой можно указать разрядность, например одно и то же число можно вывести в двоичной, десятичной, и 16 ричной системе. Вы скорее всего нескоро будете это использовать, но знать о такой возможности нужно.

С выводом все понятно, а вот с принятием данных все чуточку интереснее. Когда мы отправляем на ардуино данные, они складываются в буфер, и ждут пока ардуино их прочтает. Ксатти объём буфера составляет 64 байта. Чтобы тысячи раз в секунду не читать пустой буфер, есть функция проверки буфера, сериал аваилабл. Она возвращает число байт, которые лежат в буфере. То есть мы можем использовать условие если в буфере больше 0 байт, и получим кусок кода, который будет выполняться пока в буфере что то есть. И теперь можно попробовать принять данные. Создадим переменную типа интежер, и присвоим ей данные с порта при помощи функции чтения, сериал Рид. И сразу пошлем обратно то что получили при помощи уже известной нам сериал принтлн. То есть ардуна должна отправить нам то, что мы отправили ей. Такое эхо получается. Проверяем. Отправляю 5 и получаю 53. Отправляю 35 и получаю целых 2 числа, 51 и 53. Да, результат вообще не такой какой нам нужен. Даже близко. Смотрите в чем дело. Функция сериал рид принимает не численные значения, а символьные. А так как тип данных у нас целочисленный, мы получаем код символа согласно стандартной таблице символов. Отправим букву А. Видите, это код буквы в

таблице, номер 65. Чтобы принять символ, можно изменить тип данных на char, и тогда мы сможем отправлять и принимать символы. Но поймите, что это всё ещё символы, а не численные значения. То есть принимать и выполнять с ними математические операции нельзя, результат будет непредсказуемый.

А если мы хотим принимать именно цифры? Тогда нужно сразу переводить данные с порта в численное значение, для этого нужно вычесть символ 0, в одинарных кавычках. Это нужно просто запомнить. Теперь мы отправляем и получаем именно цифры, отлично. Полученные таким образом цифры можно использовать дальше, и в математических операциях тоже. Но к сожалению только однозначные. Число, состоящее больше чем из одной цифры, разбивается на кусочки, и получается что наша переменная принимает значение только последней цифры отправленного числа. То есть отправили 12345, и переменная перезаписывается 5 раз, и в конце концов принимает значение 5.

Для того чтобы принять все число целиком есть функция парс инт. Эта функция подождёт, пока придут все отправленные вами цифры, и сложит из них число. Никакой нолик вычитать больше не нужно. Теперь наша переменная примет большое число полностью, и можно работать с ним дальше. Но так как эта функция ждёт новые цифры, вы можете заметить небольшую задержку выполнения. Вот такая небольшая плата за удобство.

Еще есть функция флуш, она очищает буфер порта. Она пригодится в каких то жутких конструкциях работы с портом, лично я еще ни разу ей не пользовался. На этом теория работы с монитором заканчивается, но она обязательно ещё всплывёт в дальнейших уроках, о обрастёт некоторыми особенностями. Спасибо за внимание, с вами был канал заметки ардуинщика, всем пока.