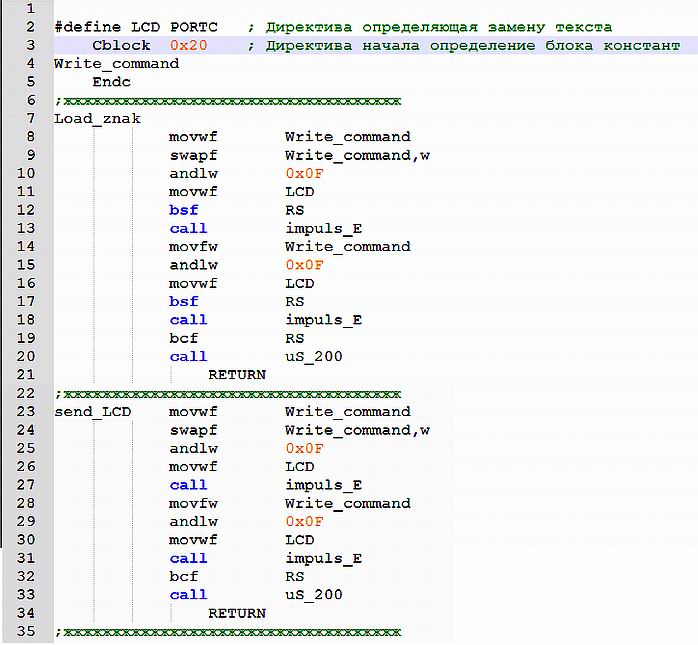
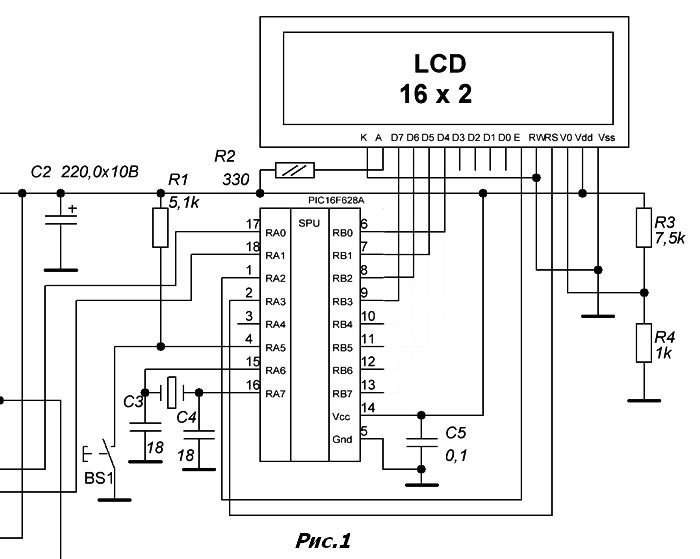
Код программы загрузки команд и символов в ЖКИ с контроллером HD44780.

Эта статья из рубрики «Программирование для начинающих» и речь в ней пойдет об одном из вариантов кода загрузки команд и символов в ЖКИ. Все будет с комментариями и объяснениями. Свои первые шаги в программировании я, как и все, начинал с изучения и анализа чужих программ. Естественно, некоторые программные решения оставлял себе. Но проходило время, и появлялись другие идеи и схемы и эти коды уже не удовлетворяли новым запросам. Так недавно случилось и с подпрограммой общения микроконтроллера с ЖКИ. Ниже вы видите один из некоторых вариантов кода, которым я пользовался много времени, пока однажды не возникли с ним проблемы. Все дело в одной инструкции. Я ни на что не претендую и уверен, что все, что я здесь опишу, было придумано давным-давно до меня, но в сети я этого не нашел.

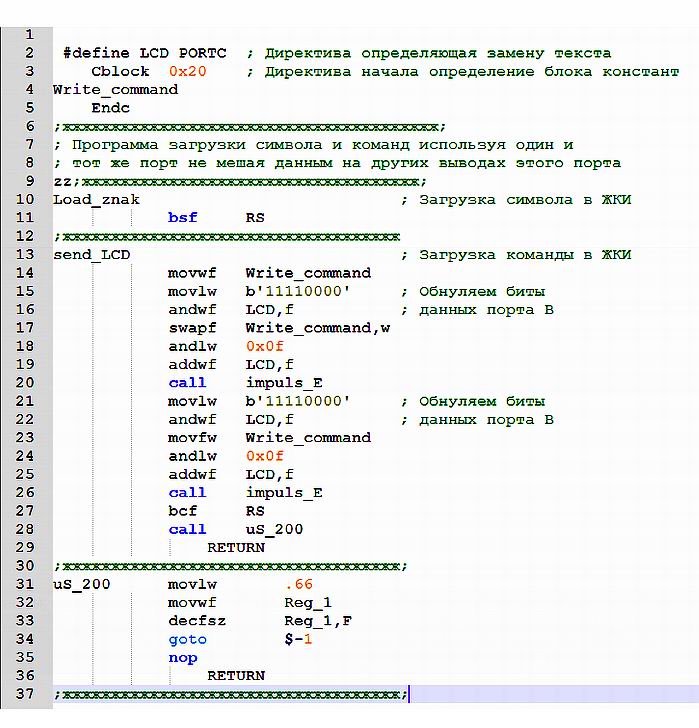


А инструкция эта находится на строке 11, 16 и 26, 30 – movwf LCD ; Запись четырех байт данных в порт. LCD – это замена текста и подразумевает один из портов микроконтроллера. Записывая всего четыре бита информации, мы такой инструкцией модифицируем все биты порта. Например, после инструкции на строке 11 - andlw 0x0F – в аккумуляторе мы будем иметь, к примеру, что-то вроде 0000 1011. И записывая это значение в порт, мы нолями в старшей тетраде, (биты 4,5,6,7) передаваемых данных в индикатор, сотрем все, что до этого

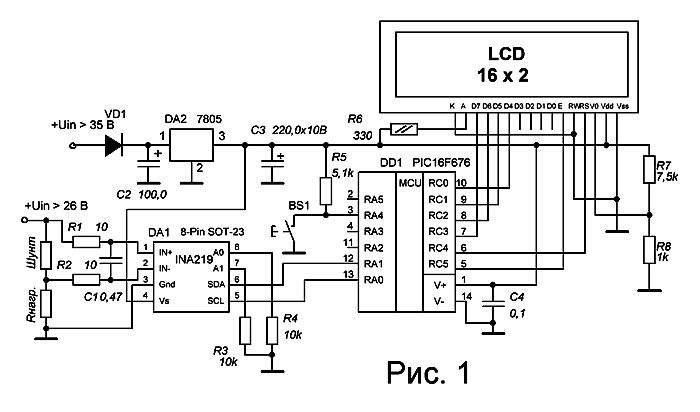
присутствовало на выходах порта – RB4, RB5, RB6, RB7. Приходилось с этим мириться или задействовать оставшиеся вывода порта как входы. Ниже приведена часть схемы с пустыми выводами.



Со временем приходил опыт, а с ним и знания, а может наоборот. И что бы исключить влияние процесса записи данных в порт на неиспользуемые выхода, я применяю теперь вот такой код. Смотрим скриншот.

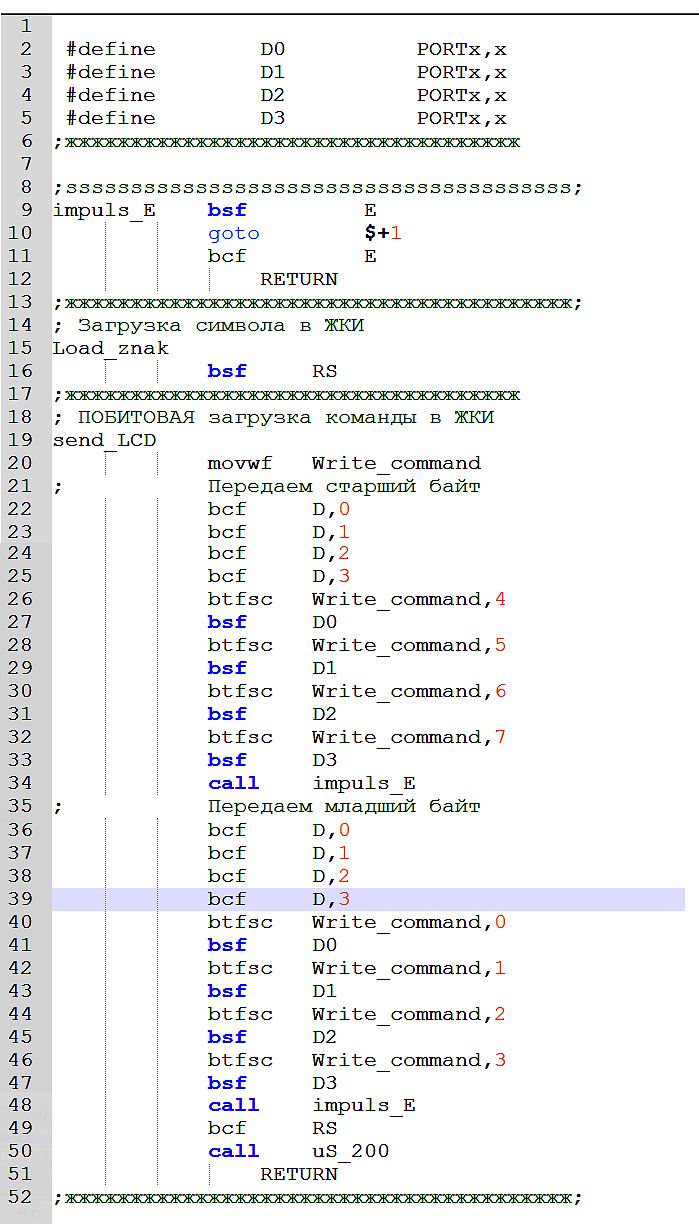


А ниже приведена схема блока измерений с использованием LCD и микроконтроллера PIC16F676. В программе для данного микроконтроллера для общения с LCD используется один порт С. Загрузка данных в порт С, посредством инструкции addwf, никоим образом не затрагивает состояние шины RS.



Этот код предполагает использование для передачи данных младшие биты порта. Начнем с передачи символа. И так, первая команда – поднимаем шину RS – строка 11. Теперь LCD в курсе, что ему будут передавать код символа, а не команды. Строка 14 – записываем нужные нам данные в регистр Write\_command для последующей передачи в индикатор. Строка 15 – заносим константу (маска)в аккумулятор WREG. И далее осуществляем логическую операцию поразрядного «И» в регистре порта, в данном случае С - andwf LCD,f - строка - 16. Таким образом, все биты, которые мы выбрали для передачи в ЖКИ, будут обнулены, не затрагивая значения не используемых бит порта. Выделяем для передачи старшие биты данных из регистра Write\_command (строки 17,18) и оставляем их в рабочем регистре. Далее идет интересная инструкция, не записи в порт, а прибавление наших данных к данным уже имеющимся в регистре порта, строка 15 – addwf LCD,f. Таким образом, наши младшие биты данных займут место в младших битах порта. А прибавленные нули в старшей тетраде к старшей тетраде регистра порта информацию в ней не изменят. Строка 20 – вызов подпрограммы формирования стробирующего импульса. Это мы передали в индикатор старшие биты данных хранящихся в регистре Write\_command. Далее точно так же передаются и младшие биты. В конце передачи не забываем прижать шину RS, строка 27. Все просто… , когда знаешь.

Есть еще один код, это уж когда вообще остались свободными одиночные биты в разных портах. Это код подпрограммы с побитовой записью данных. Он занимает больше места в памяти программ, чем предыдущий код. Смотрим скриншот ниже.



Строки 2…5 – замена текста. Присваиваем каждому выбранному биту портов свои названия для удобства. Строки 9… 12 – подпрограмма формирования стробирующего импульса на шине Е. Поднимаем шину RS – сообщаем контроллеру индикатора, что сейчас будет загрузка символа - bsf RS. Переносим данные из аккумулятора в регистр для загрузки, строка 20. Обнуляем задействованные выводы портов или порта, строка 22…25. Далее идет побитовая передача данных старшей тетрады регистра Write\_command в выбранные биты портов. Строка 26 – проверяем состояние 4-го бита регистра загрузки, если «1», то и в вывод порта D0 записываем единицу. Если ноль, то следующая инструкция игнорируется и на выводе D0 останется ноль (этот бит был обнулен заранее строка 22), переходим к пятому биту. Так же анализируем и переносим его значение на вывод порта – D1 и т.д. и т.п. В конце кода загрузки прижимаем шину RS и ждем 200мкс, когда контроллер индикатора усвоит переданные данные.