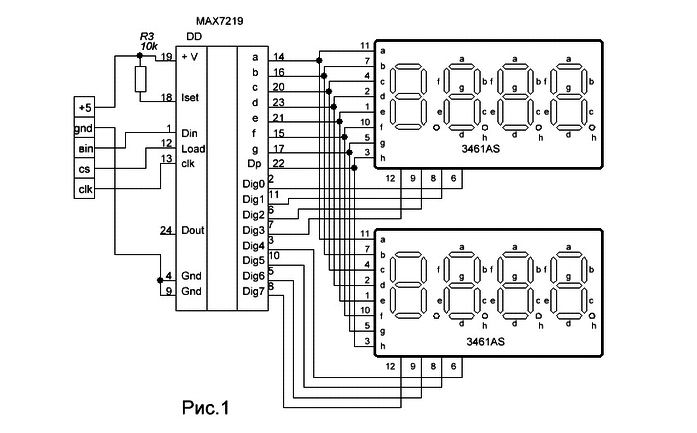
Взаимодействие PIC контроллеров с драйвером светодиодных индикаторов MAX7219

Во многих приборах и устройствах для отображения информации используют светодиоды или семисегментные светодиодные индикаторы. Достоинством подобных устройств отображения информации является наличие большой яркости и контрастности. Они хорошо различимы под любым углом при любом освещении. Так же светодиоды имеют меньшую восприимчивость к механическим нагрузкам, чем ЖКИ. Многие светодиоды могут работать в области отрицательных температур. При соблюдении соответствующих режимов срок службы может достигать 50 000 часов. Но практически все светодиодные семисегментные индикаторы и светодиодные матрицы работают в режим динамической индикации, который применяют для построения многоразрядных индикаторов. При таком режиме сегменты и разряды индикатора работают не одновременно, а по очереди. Переключение происходит с большой скоростью, из-за этого человеческий глаз не замечает этого переключения. Так как у светодиодов очень малая инерционность, сменяющиеся разряды сливаются в одно изображение.

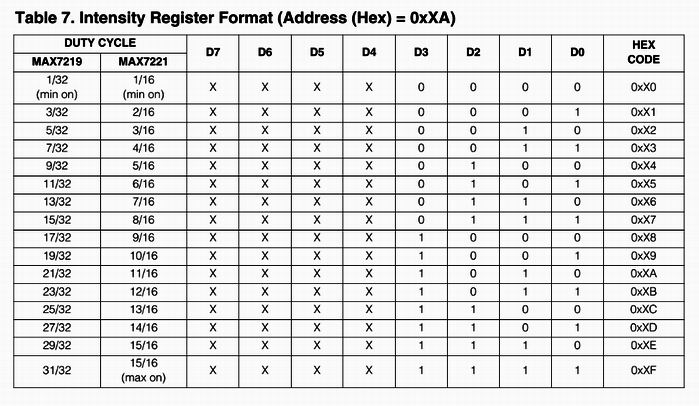
Для упрощения кода программы индикации и сбережения ресурсов микроконтроллера в схемах применяют драйвер MAX7219. Микросхема MAX7219 предназначена для управления семисегментными светодиодными индикаторами и светодиодными матрицами. При этом отпадает необходимость в реализации динамической индикации. Управление драйвером осуществляется по интерфейсу SPI, для реализации которого потребуется выделить всего 3 линии ввода/вывода микроконтроллера, линия тактовых импульсов - CLK, DIN – линия данных и CS - последовательные данные загружаются в сдвиговый регистр, пока CS низкий. После передачи 16 бит данных они защелкиваются передним фронте CS. Данные передаются словом в 16 бит. Максимальная частота тактирования равна 10 МГц. Это позволяет при частоте опорного генератора микроконтроллера 4мГц не вводить в код программы взаимодействия с этим драйвером подпрограмм задержек. Внешний вид китайского модуля показан на фото 1.



Здесь 4х разрядные индикаторы расположены последовательно, что удобно при работе с большими числами (например, индикаторы частотомеров). На сайте есть статья» «[Индикатор восьмиразрядный на MAX7219](https://www.kondratev-v.ru/indikatory/indikator-vosmirazryadnyj-na-max7219.html" \o "Индикатор восьмиразрядный на MAX7219)» о модуле с вертикальным расположением индикаторов. На рисунке 1 представлена схема модуля.



Модуль MAX7219 позволяют регулировать яркость дисплея с помощью внешнего резистора (RSET), подключенного между V+ и ISET. Пиковый ток, подаваемый сегментными драйверами, номинально в 100 раз превышает ток, подаваемый на ISET. Яркость дисплея также можно регулировать цифровым способом с помощью регистра яркости. Цифровое управление яркостью дисплея осуществляется с помощью встроенного широтно-импульсного модулятора, который управляется нижней строкой регистра интенсивности. Модулятор масштабирует средний ток сегмента за 16 шагов от максимального значения 31/32 до 1/32 от пикового тока, установленного RSET. В таблице 7 приведен формат регистра яркости свечения DP. Минимальное время заполнения междузначных разрядов установлено равным 1/32 цикла.



Для управления микросхемой требуется вначале записать данные в управляющие регистры, провести начальную настройку режимов работы драйвера, т.е. ее инициализацию.

Прежде всего, переводим драйвер MAX7219 из тестового режима, когда засвечиваются все сегменты индикаторов, в нормальный режим работы. В регистр с адресом 0x0F данные, в формате b'xxxxxxx0'. Ноль в младшем бите переводит драйвер в нормальный режим работы.

<code>

movlw 0x0F ; Выключить тестовый режим

movwf Adres\_7219

movlw 0x00

movwf Data\_7219

call TX\_7219

</code>

Далее записываем в регистр с адресом 0x0С код команды вывода из спящего режима – 0х01.

<code>

movlw 0x0С

movwf Adres\_7219

movlw 0х01

movwf Data\_7219

call TX\_7219

`</code>

Установка яркости. Учитывая страну происхождения всех приобретаемых индикаторов, советую не выставлять большую яркость сегментов индикаторов, долго не наработают. Есть горький опыт. Адрес регистра, в который заносятся данные о яркости – 0х0А. Я выбрал число 3 и 15 уровней яркости, но это при условии, что общий задатчик тока сегментов – R3 на схеме 1, имеет сопротивление 10к. Это резистор, задающий максимальный ток через сегменты индикатора, минимальное сопротивление резистора равно 9,53 кОм, при этом максимальный ток составит 40 мА. Его можно заменить на потенциометр и вывести на переднюю панель прибора или другого устройства для ручной регулировки яркости свечения индикаторов. Или заменить на фоторезистор и тогда можно получить автоматическую регулировку яркости индицирования выводимой информации. Для экономии энергии используйте R3 = 47 кОм при использовании внешних источников питания индикаторов.

<code>

movlw 0x0a

movwf Adres\_7219

movlw .3 ; Значение яркости можно выбрать от 0... 15.

movwf Data\_7219 ; 15 - максимальная яркость.

call TX\_7219

</code>

Драйвер MAX7219 позволяет устанавливать количество используемых разрядов индикатора в зависимости от ваших нужд на данный момент. Адрес регистра – 0х0b.

<code>

movlw 0x0b

movwf Adres\_7219

movlw b’00000111’ ; 0 – включен 1разряд .... 7 – включено все 8 разрядов.

movwf Data\_7219

call TX\_7219

</code>

MAX7219 имеет еще одну замечательную опцию – декодирование, т.е. мы можем подавать на вход данные о числах в двоично-десятичном коде. Это упрощает программу и высвобождает определенные ресурсы памяти программ микроконтроллера. Драйвер сам перекодирует их в семисегментный код. Эту опцию можно включать и выключать отдельно для каждого разряда. Декодирование выключают, если вместо цифр на индикатор необходимо вывести буквы и символы. Например, если в регистр декодирования записать b'00000000' – все разряды будут без декодирования, b'11111111' – все разряды с декодированием и ’01100110’ – 0,3,4,7 разряды без декодирования, в них запишем нужные символы «-» и «°», а 1,2,5,6 с декодированием. При выводе, например, температуры от двух датчиков увидим -15° -30°. При включенном декодировании декодер анализирует только младшую тетраду (младший полубайт) регистра данных. Данные в старшей тетраде игнорируются, но седьмой бит регистра это десятичная точка. Поэтому в регистр с двоично-десятичным кодом цифры, после которой надо поставить точку, надо дописать «1» в седьмой бит. Пример – «9.», код в регистре – b'1000 1001’.

<code>

movlw 09h

movwf Adres\_7219

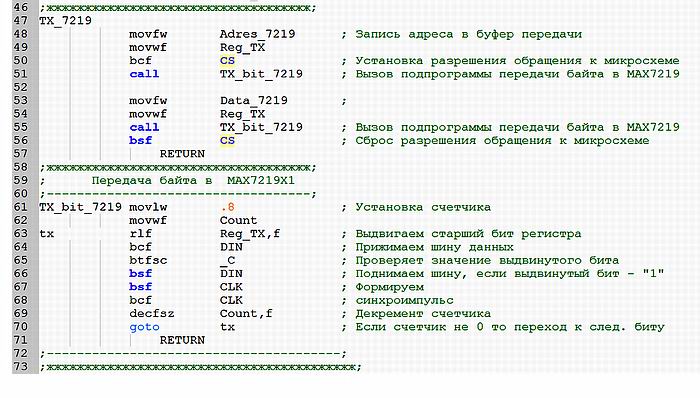
movlw 0 ; Все восемь разрядов без декодирования.

movwf Data\_7219

call TX\_7219

</code>

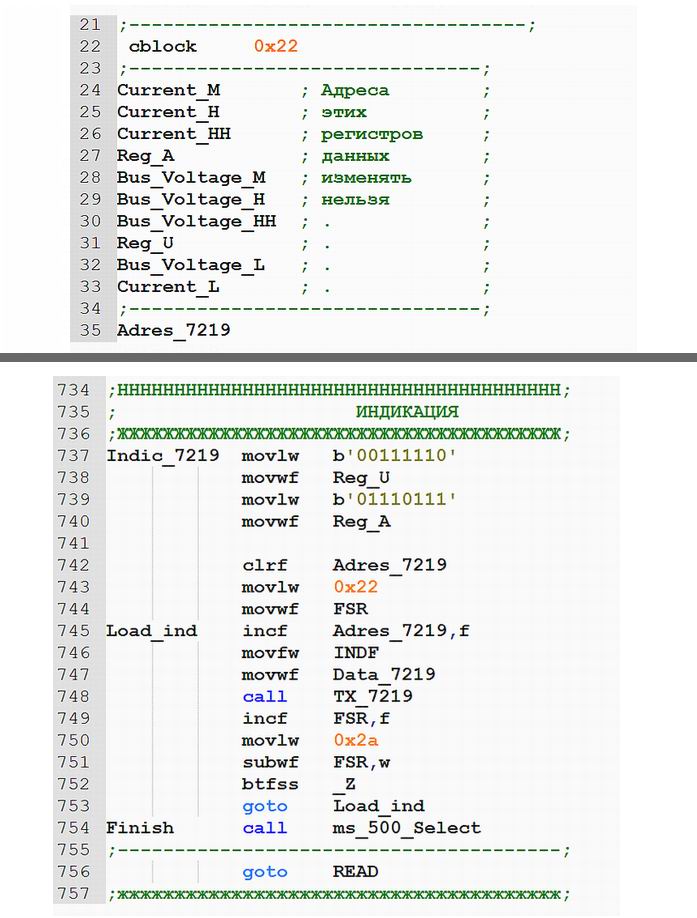
С инициализацией закончили, теперь о подпрограмме загрузке данных в драйвер. Ниже приведен скриншот с кодом передачи в MAX7219.



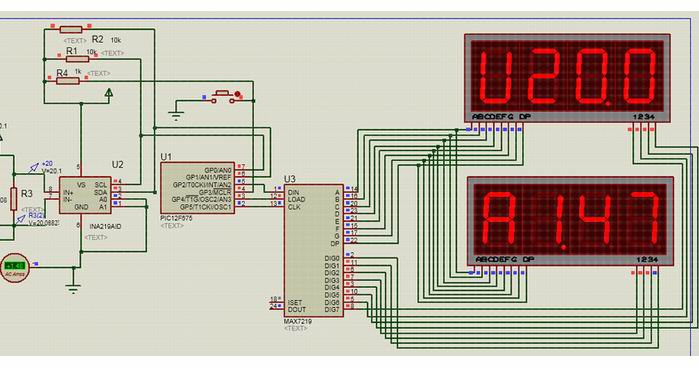
Вывод данных на индикацию

Для передачи данных в драйвер лучше использовать косвенную адресацию. При этом надо неукоснительно выполнять условие – у регистров, хранящих данные для вывода на индикатор, нельзя менять адреса.

На скриншоте ниже показан код вывода данных на индикатор и размещение необходимых регистров в блоке.

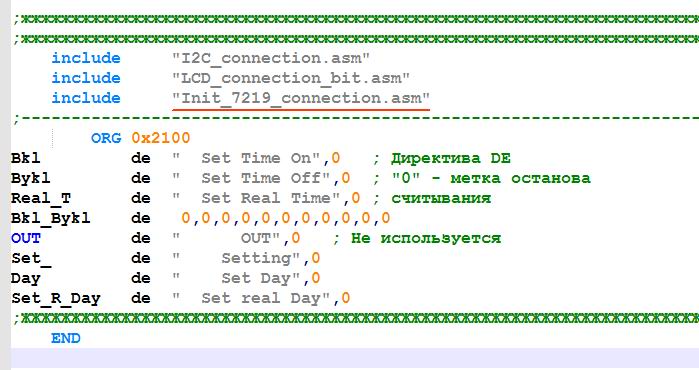


А вот и результат в PROTEUS



В архиве со статьей будет подсоединяемый файл –«Init\_7219\_connection.asm». В нем программа инициализации драйвера MAX7219. Для подсоединения данного файла к основному исходнику надо в конце основного файла перед «END», записать строку с директивой «include».

Примерно так: include "Init\_7219\_connection.asm" См. пример ниже:



Программа инициализации начинается с метки Init\_MAX7219. Поэтому чтобы обратиться к этому подсоединяемому файлу используйте инструкцию CALL – вызов подпрограммы – CALL Init\_MAX7219. После исполнения инструкций инициализации активная точка программы по стеку вернется на следующую инструкцию основного исходного файла. Сам файл «Init\_7219\_connection.asm» должен находиться в папке проекта вместе с исходным файлом программы. Это очень удобно, не надо переписывать лишний раз код. На этом все. Удачи. К.В.Ю.