Программирование пользовательских символов

Символьный дисплей выполнен на контроллере HD44780 позволяет загружать до 8 пользовательских символов любого начертания

Для программирования своего символа требуется загрузить адрес символа, его графическое представление – 8 байт. В котором последние пять бит будут отображать пиксели в 8 строках на символ. Установленный бит - затемненный пиксель. После программирования всех необходимых символов требуется заново инициализировать дисплей. Сам вывод можно производить командой LCDD х , где х – код символа от 0 до 7. Или LPRINT $(х), где переменная $(х) содержит код символа от 1 до 7.

Собственно пример программы

*REM '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*REM ' Подпрограмма программирования нового символа*

*REM ' для дисплея HD44780*

*REM ' VER 4.0*

*REM '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*REM ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*REM ' Требуется ОБРАБОТКА ПРЕКОМПИЛЯТОРОМ BASIC*

*REM '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*0000*

*CLR:LINIT 1*

*REM ' Любой символ имеет адрес расположения равный 0х00 - 0х07*

*REM ' мы можем записать до 8 необходимых нам символов*

*REM ' Символ состоит из матрицы 5\*8 точек, как праавило*

*REM ' нижняя строка не занята, для курсора*

*REM ' пример термометра*

*REM '*

*REM ' \* \* # \* \* 0b0000100 = 4*

*REM ' \* # \* # \* 0b0001010 = 10*

*REM ' \* # \* # \* 0b0001010 = 10*

*REM ' \* # # # \* 0b0001110 = 14*

*REM ' \* # # # \* 0b0001110 = 14*

*REM ' # # # # # 0b0011111 = 31*

*REM ' # # # # # 0b0011111 = 31*

*REM ' \* # # # \* 0b0001110 = 14*

*REM '*

*REM ' пример Капельки*

*REM '*

*REM ' \* \* # \* \* 0b0000100 = 4*

*REM ' \* \* # \* \* 0b0000100 = 4*

*REM ' \* # \* # \* 0b0001010 = 10*

*REM ' \* # \* # \* 0b0001010 = 10*

*REM ' # \* \* \* # 0b0010001 = 17*

*REM ' # \* \* \* # 0b0010001 = 17*

*REM ' # \* \* \* # 0b0010001 = 17*

*REM ' \* # # # \* 0b0001110 = 14*

*REM '*

*REM ' Еденичками кодируются закрашенный точки*

*REM ' Для записи требуется*

*REM ' Выбрать адрес символа*

*REM ' Записать собственно битовую таблицу*

*[New\_symbol]*

*REM ' это будет символ с кодом ноль*

*REM ' для записи код надо умножить на 8 и + 64,*

*REM 'Запишем его, формируем команду*

*LCDC 64:REM ' код =0; = 0\*8+64*

*REM ' Определим таблицу для символа*

*DATA #(10),4,10,10,14,14,31,31,14*

*REM ' Грузим символ - можно вынести в подпрограмму*

*FOR i=10 TO 17*

*LCDD #(i)*

*NEXT i*

*LCDC 72:REM ' код =1; = 1\*8+64*

*REM ' Определим таблицу для второго символа*

*DATA #(10),4,4,10,10,17,17,17,14*

*REM ' Грузим символ*

*FOR i=10 TO 17*

*LCDD #(i)*

*NEXT i*

*REM 'Заново настроим дисплей, и все будет готово*

*LINIT 1*

*REM 'Печать символов*

*AT 0*

*LCDD 0:LCDD 1*

*REM 'Для кодов , отличных от нуля можно использовать LPRINT*

*AT 5*

*#(1)=1*

*LPRINT $(1)*

*END*

Программа после компилятора

*0000 CLR:LINIT 1*

*0001 LCDC 64:REM ' код =0; = 0\*8+64*

*0002 DATA #(10),4,10,10,14,14,31,31,14*

*0003 FOR i=10 TO 17*

*0004 LCDD #(i)*

*0005 NEXT i*

*0006 LCDC 72:REM ' код =1; = 1\*8+64*

*0007 DATA #(10),4,4,10,10,17,17,17,14*

*0008 FOR i=10 TO 17*

*0009 LCDD #(i)*

*0010 NEXT i*

*0011 LINIT 1*

*0012 AT 0*

*0013 LCDD 0:LCDD 1*

*0014 AT 5*

*0015 #(1)=1*

*0016 LPRINT $(1)*

*0017 END*

Результат выполнения

