**Работа 4 (часть 2)**

**"Вывод текстовой информации на дисплей МЭЛТ МТ-12864J"**

**Цель работы:** закрепить основы работы с дисплеем МЭЛТ МТ-12864J.

**Задачи:**

– закрепить на практике особенности ЖК дисплея МЭЛТ МТ-12864J;

– закрепить на практике особенности подключения сторонних библиотек к проекту;

– изучить особенности вывода символьной информации на дисплей МТ-12864J;

– изучить особенности вывода графической информации на дисплей МТ-12864J;

– получить навыки проектирования пользовательского графического интерфейса для микроконтроллера с использованием жидкокристаллического индикатора (ЖКИ);

– закрепить навыки применения библиотек периферийных устройств МК на примере библиотек RST\_CLK и PORT;

– закрепить навыки программирования микроконтроллеров на языке C.

**Используемое оборудование:**

1. Отладочная плата с микроконтроллером Миландр MDR1986BE92QI/MDR1986BE93У.

2. Комплект Программатора

2.1 программатор JLINK (USB-JTAG);

2.2 Кабель USB 2.0 A – B;

2.3 Шина (20 проводников).

3. Источник питания 5В (или дополнительный USB кабель)

4. Комплект Логического анализатора

4.1 Логический анализатор

4.2. Кабель USB 2.0A – mini-B

4.3 Шина 10 проводников.

**Используемая документация:**

1. Техническое описание на ядро Миландр MDR1986BE9x (файл «1\_Тех\_описание\_ядро\_1986ВЕ9X.pdf или по ссылке с сайта разработчика [https://ic.milandr.ru/upload/iblock/a33/ioaf9ygfmq1lbxfhd5aad0mukg3dc93s/1986%D0%92%D0%959X.pdf](https://ic.milandr.ru/upload/iblock/a33/ioaf9ygfmq1lbxfhd5aad0mukg3dc93s/1986ВЕ9X.pdf))

2. Выводы отладочной платы микроконтроллера 1986BE92QI (файл «2\_Выводы\_платы\_1986BE92QI.pdf или по ссылке\* с сайта разработчика <https://ic.milandr.ru/upload/iblock/8f6/8f67b8b736b3ec94edbbeb4777a9c4db.zip>)

3.Выводы отладочной платы микроконтроллера 1986BE93 (файл «2\_Выводы\_платы\_1986BE93У.pdf или по ссылке\* с сайта разработчика <https://ic.milandr.ru/upload/iblock/782/782c4c3b486d6f8d92995e9a44a94401.zip>)

При загрузке схемотехнической документации с сайта Milandr (информация о выводах платы), загружается zip архив, который содержит 1 или 2 файла со схемой размещения выводов микроконтроллера на плате и несколько схемотехнических файлов для разводки и печати платы.

**Библиотечные функции дисплея для вывода текста**

В файлах mlt\_lcd.c и mlt\_lcd.h представлены функции для работы с дисплеем МЭЛТ МТ-12864J. Все они уже описаны в предыдущей части работы. Данная часть посвящена функциям для вывода текста (таблица 4.1):

Таблица 4.1 – Функции для вывода

|  |  |
| --- | --- |
| **Подпрограмма** | **Описание** |
| void LcdPutChar (uint8\_t\* array, int Xpos, int Ypos) | Вывод символа |
| void LcdPutString (uint8\_t\*\* array, int Ypos) | Вывод строки |
| int LcdScrollString (uint8\_t\*\* array, int Ypos, int size,int cnt) | Сдвиг строки |
| void LcdPutImage (uint8\_t\* array, int Ypos1, int Xpos1,int Ypos2, int Xpos2) | Вывод изображения |

**Файл кодировки**

Дисплей представляет собой пиксельную матрицу, и любой символ, изображаемый на нем, должен быть прорисован в виде набора тёмных пикселей.

Сам дисплей не знает кодировок и выводить символы самостоятельно не умеет. В библиотеке mlt\_lcd описаны функции вывода символов, в которых подразумевается, что один символ – это массив из 8-ми значений, описывающих темные точки на дисплее (каждый символ 8x8 пикселей, которые лежат на одной «странице» дисплея).

Описание массивов значений для прорисовки символов (таблица кодировки) представлено в файле font.h. В сети Интернет есть множество сервисов, позволяющих сформировать шрифт для LCD дисплеев (например, <http://www.mikroe.com/glcd-font-creator/>).

**Стандартные функции обработки символьной информации языка C. Sprintf**

Функция sprintf определена в заголовке stdio.h и предназначена для записи в строковый буфер (массив символов) строки в определенном формате.

В общем виде вызов функции выглядит следующим образом:

sprintf(char\* outstring, const char\* part1string, const char\* part2string, …), где outstring – формируемая выходная строка;

part1string, part2string – части формируемой строки.

На первый взгляд данная функция реализует операцию конкатенации (объединения строк), но в данной функции есть возможность использовать в качестве входных параметров числовые типы данных и выводить результат в выходную строку в определенном формате. Для управления форматом вывода используются следующие спецификаторы, представленные в таблице 4.2.

Таблица 4.2 – Спецификаторы функции sprintf

|  |  |
| --- | --- |
| **Спецификатор** | **Назначение** |
| % | идентификатор преобразования типов |
| с | одиночный символ char |
| s | строка символов |
| d (или i) | преобразует знаковое целое число в набор символов |
| o | преобразует беззнаковое целое в набор символов в 8ой системе счисления |
| x (или X) | преобразует беззнаковое целое в набор символов в 16й системе счисления  если x, то a,b,c,d,e,f; если X, то A,B,C,D,E,F |
| u | преобразует беззнаковое целое число в набор символов |
| f (или F) | преобразует число в формате с плавающей точкой в десятичную запись в формате [-]dddd.dddd |
| e (или E) | преобразует число в формате с плавающей точкой в десятичную запись в экспоненциальной форме в формате [-]d.ddd e+/-dd, если использовать спецификатор E, то при выводе будет прорисована заглавная буква E |
| a (или A)  (С99) | преобразует число в формате с плавающей точкой в десятичную запись в экспоненциальной форме в формате [-]0xh.hhh p+/-d, если использовать спецификатор A, то при выводе будут прорисованы заглавные символы |
| g (или G) | преобразует число в десятичной форме с плавающей точкой или десятичной экспоненциальной форме в десятичную запись с определенной точностью, если использовать спецификатор G, то при выводе будут прорисованы заглавные символы |
| n | возвращает количество символов в записи |
| p | возвращает последовательность символов по указателю |

**Практическая часть**

**Задание 1**. Создайте новый проект аналогично заданию 1 из первой части работы. Дополнительно добавьте к проекту файл lcdstring.c аналогично файлу mlt\_lcd.c. Убедитесь, что файлы stdio.h, lcdstring.h, mlt\_lcd.h, font.h расположены в директории LIBS в проекте и подключите эти заголовочные файлы с использованием директивы #include к файлу main.c. Добавьте функцию Delay и исправьте функцию main в соответствии с примером, представленным ниже.

|  |
| --- |
| void Delay(uint32\_t wait)  {  for (int i=0;i<wait;i++);  }  int main (void) //точка входа в программу  {  CPUinit(); //настройка тактирования процессора  PCLKinit(); //включние тактовых сигналов  LCDPins(); //инициализация выводов  LCDStart(); //запуск LCD  LcdClearChip (1);  LcdClearChip (2);  LcdPutImage(icon\_mil,0,0,1,1);  LcdPutChar (cyr\_R, 11,0);  LcdPutChar (cyr\_G, 12,0);  LcdPutChar (cyr\_R, 13,0);  LcdPutChar (cyr\_T, 14,0);  LcdPutChar (cyr\_U, 15,0);  uint8\_t \*group[3] = {{dig\_9},{dig\_4},{dig\_5}};  LcdScrollString (group, 2, 3, 8);  uint8\_t \*str[16]={{cyr\_V},{cyr\_R},  {cyr\_A},  {cyr\_G},  {cyr\_I},{sym\_sp},{sym\_sp},{sym\_sp},{sym\_sp},{sym\_sp},{sym\_sp},{sym\_sp},{sym\_sp},{sym\_sp},{sym\_sp},{sym\_sp}  };  char stroka[16];  sprintf(stroka, "%s %f", "Pi ",3.141592);  PrintString (stroka,4);  while(1) //бесконечный цикл  {  for (int i=0;i<16;i++)  {LcdScrollString (str, 3, 5, i);  Delay(1000000);}  }  } |

Выполните сборку программы, загрузите программу в контроллер. Проанализируйте полученный результат. Проанализируйте выполнение функций LcdPutChar(), LcdPutString(), LcdScrollString(), LcdPutImage. Изучите код функции PrintString() и содержимое файла font.h. В отчёте представьте выводы, полученные по результатам проведенных исследований.

**Задание 2**. Измените программу из задания 1 таким образом, чтобы в нижнем правом углу экрана добавился номер бригады и фамилии студентов.

Выполните сборку программы, загрузите программу в контроллер. Проанализируйте полученный результат. Зафиксируйте результат в отчёте.

**Задание 3**. Измените программу из задания 1 таким образом, чтобы на экране отображалось меню в следующем виде:

|  |
| --- |
| МК1986ВЕ9х  Задача 1   Задача 2  № бригады  ФИО1  ФИО2 |

Вместо символа «<» используйте курсор из файла font.h. Подключите к проекту кнопки UP и DOWN и реализуйте переходы курсора по меню с использованием этих кнопок.

Выполните сборку программы, загрузите программу в контроллер. Проанализируйте полученный результат. Зафиксируйте результат в отчёте.

**Задание 4**. Измените программу из задания 3, добавив на каждый пункт меню свою обрабатываемую задачу:

Задача 1 – изменение состояния вывода PORTA6 (PORTC0 для 1986BE93У) на противоположное через 1 секунду с подтверждением текущего состояния на дисплее;

Задача 2 – изменение состояния PORTA7 (PORTF5 для 1986BE93У) на противоположное через 2 секунды с подтверждением текущего состояния на дисплее.

Выполните сборку программы, загрузите программу в контроллер. Подключите логический анализатор к выводам порта А (портов C и F для 1986ВЕ93У) (см. работу 1) Проанализируйте полученный результат. Зафиксируйте результат в отчёте.

**Вопросы для самоконтроля**

1. Зачем нужен файл font.h?

2. Какой размер в пикселях имеет один символ при использовании font.h?

3. Поясните назначение и параметры функций mlt\_lcd, используемых для вывода текста.

4.Как можно реализовать бегущую строку на дисплее МЭЛТ МТ-12864J?

5. Как используется функция sprintf в рамках данной работы?

6. Опишите основные спецификаторы sprintf и приведите свои примеры использования функции.

7. Поясните назначение и алгоритм работы функции PrintString().