

Для начала работы с отладочной платой LDM-МСр0411100101-Q208 необходимо наличие следующих компонентов:

- операционная система Windows (в данном примере рассматривается ОС Windows, также пользователи могут работать в среде Linux);
- отладочный комплект [LDM-МСр0411100101-Q208](#);
- инструментальный программный пакет Windows (загрузить в разделе «Поддержка», [«Техническая документация и ПО»](#) на официальном сайте www.multiclet.com);
- драйвер для микросхемы FTDI , загрузить с <http://www.ftdichip.com> (если у вас Windows 7 и выше драйвер установится автоматически);
- примеры программ (загрузить в разделе «Поддержка», [«Техническая документация и ПО»](#) на официальном сайте www.multiclet.com).

1) Установите инструментальный программный пакет. По умолчанию будет выбран путь C:\MultiClet (подробную инструкцию по установке инструментального программного пакета см. в [руководстве по эксплуатации отладочной платы](#) - стр. 17)

2) Подключите mini-usb кабель к разъему 15 на отладочной плате.



Рис 1. Кабель mini-USB

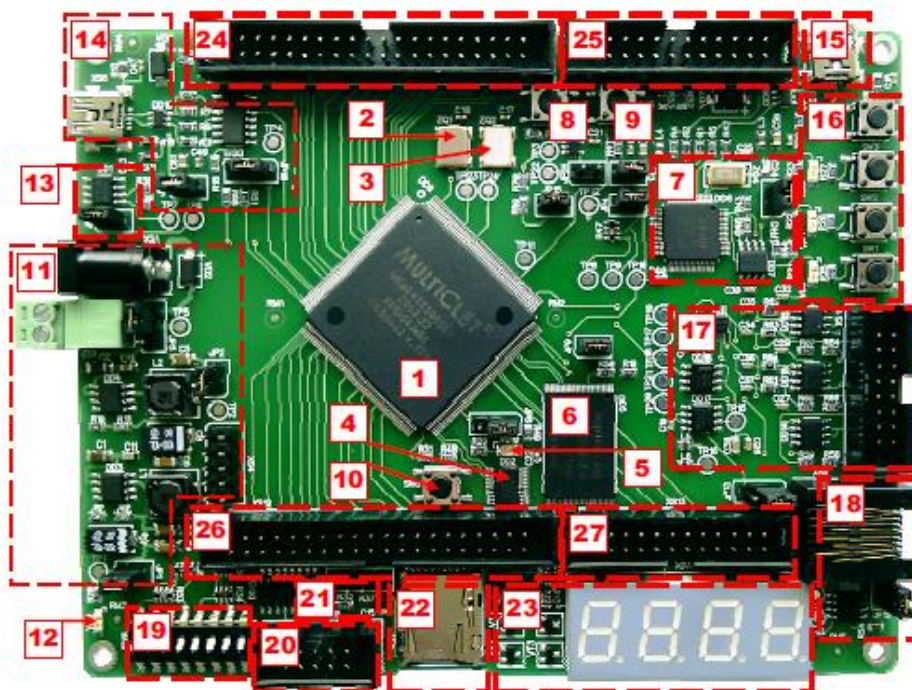


Рис 2. Отладочная плата LDM с
мультиклеточным процессором МСр0411100101

4) Установите драйвер FTDI. Если вы используете Windows 7, то драйвер для FTDI установится автоматически, в противном случае необходимо установить драйвер для FTDI вручную (установка драйвера осуществляется со снятым джампером JP7, см. пункт 7). Затем подключите кабель mini-USB к разъему USB вашего ПК.

5) Запустите пример из папки Examples (LDM-МСр04/led/led_test.ppr)

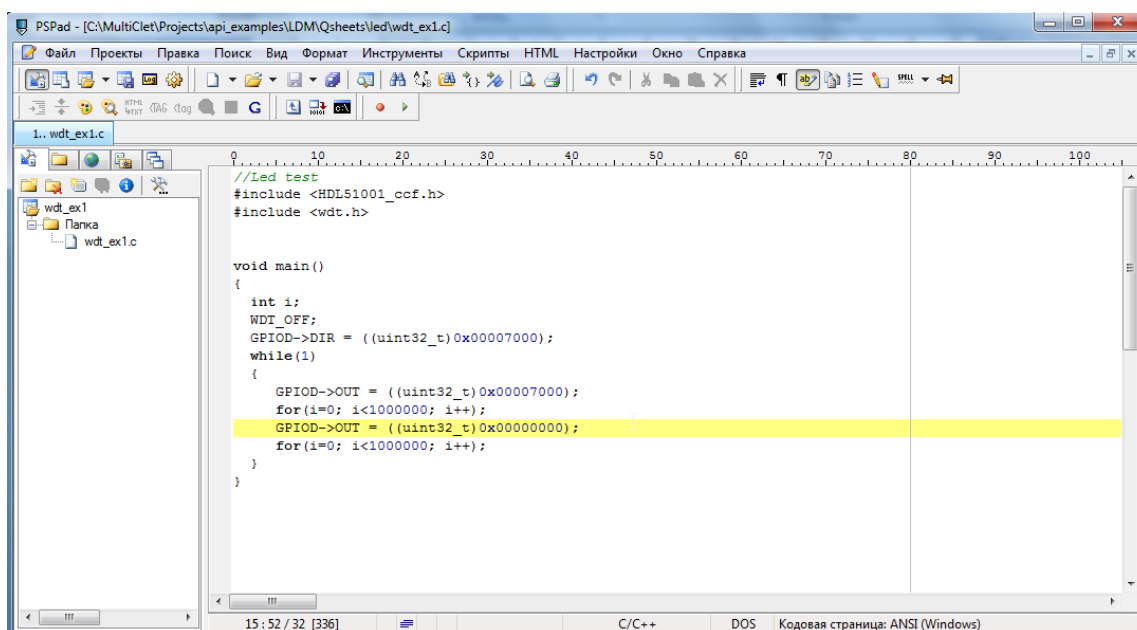


Рис 3. Окно PSPad с примером «мигание светодиодов»

Рассмотрим код программы на Си подробнее:

```
1) #include <HDL51001_ccf.h>
2) #include <wdt.h>
3) void main()
4) {
5)     int i;
6)     WDT_OFF;
7)     GPIOD->DIR = ((uint32_t)0x00007000);
8)     while(1)
9)     {
10)         GPIOD->OUT = ((uint32_t)0x00007000);
11)         for(i=0; i<1000000; i++);
12)         GPIOD->OUT = ((uint32_t)0x00000000);
13)         for(i=0; i<1000000; i++);
14)     }
15) }
```

/*

В данном примере в строках №1,2 подключена обязательная библиотека HDL51001_ccf.h и библиотека сторожевого таймера wdt.h;

- в строке №3 начинается основная программа
- в строке №5 объявляется переменная типа int (в версии процессора MCp0411100101 тип short отсутствует)
- в строке №6 использован макрос для отключения сторожевого таймера (если сторожевой таймер не отключить, то будет происходить сброс процессора каждые 53 с.)
- в строке №7 происходит настройка выводов порта D, соответствующих светодиодам 1,2,3; в документации на отладочную плату LDM на странице 36 можно ознакомиться с соответствием выводов процессора различным устройствам (в нашем случае светодиодам). Из рис.4 определяем, что светодиоды расположены на порту D, выводы 12,13,14,15, отсюда следует шестнадцатеричное представление 0x00007000 для конфигурации выводов процессора, соответствующих светодиодам 12,13,14, на выход. Сочетание uint32_t это typedef unsigned int uint32_t из библиотеки HDL51001_ccf.h
- в строке №8 начинается бесконечный цикл while()
- в строке №10 выставлена логическая единица на соответствующие выводы светодиодов 12,13,14, что включает три светодиода
- строка №11 представляет собой цикл задержки
- в строке №12 выставлен логический ноль на соответствующие выводы светодиодов 12,13,14, что выключает три светодиода
- строка №13 аналогична строке №11

*/

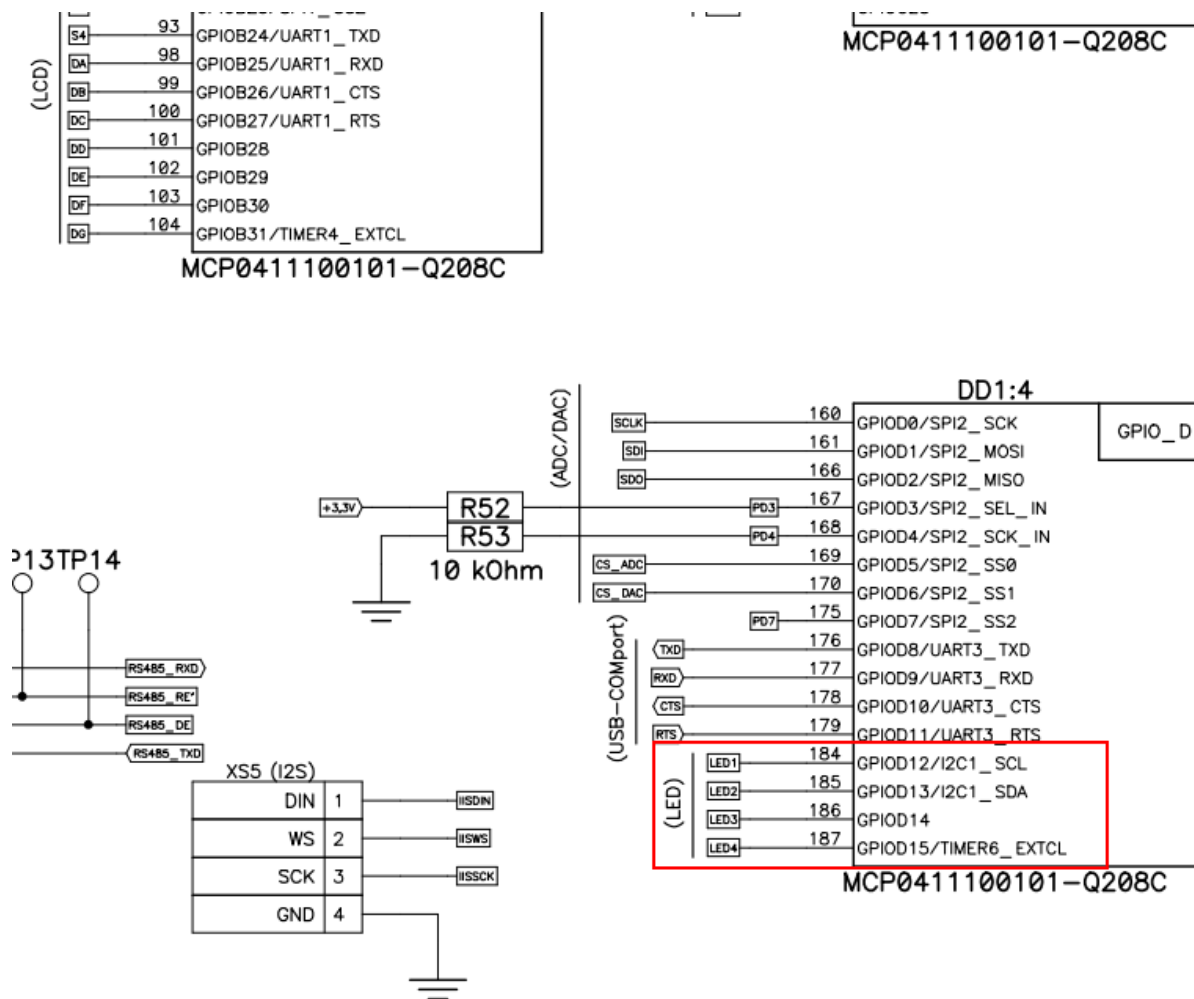


Рис 4. Выдержка из документации на отладочную плату LDM(стр. 36)

б) Скомпилируйте, рассмотренный выше пример, для этого нажмите сочетание клавиш **Ctrl + F9** или кликните левой клавишей мышки по соответствующей пиктограмме, см. рис.5.

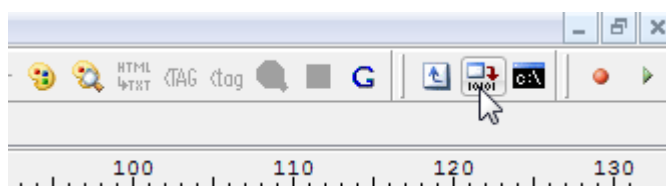


Рис 5. Компиляция проекта в PSPad

После компиляции появится окно с результатами компиляции, если вы получили сообщение “Process completed, Exit Code 0”, то компиляция прошла успешно. В случае отсутствия данного сообщения необходимо исправить ошибки в коде программы, на которые укажет компилятор. Результат компиляции приведён на рис.6.

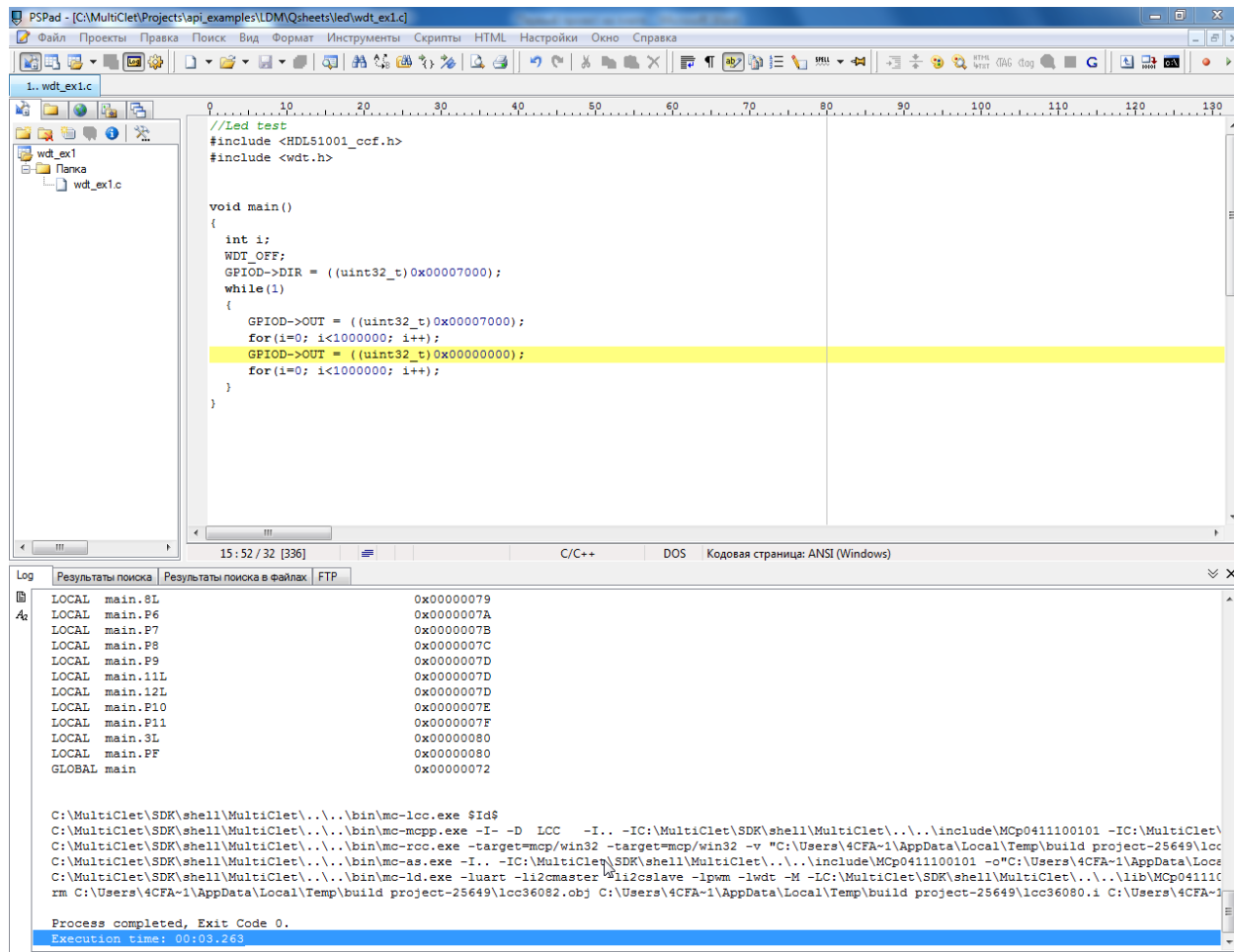


Рис 6. Окно PSPad с результатом компиляции проекта

7) Перед загрузкой проекта на отладочную плату необходимо проверить начальное положение джамперов согласно документации на отладочную плату LDM-МCp0411100101-Q208 Evolution. Теперь внесём изменения:

- джампер JP5 необходимо переключить в положение – “питание от USB”, см. рис. 7.
- джампер JP7 должен отсутствовать, см. рис. 8.



Рис 7. Положение джампера JP5



Рис. 8. Джампер JP7 отсутствует

8) Для загрузки проекта на плату необходимо нажать сочетание клавиш **ALT + F9** или кликнуть левой клавишей мышки по соответствующей пиктограмме, см. рис. 9.

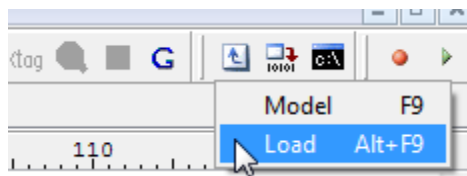


Рис. 9. Загрузка программы на плату

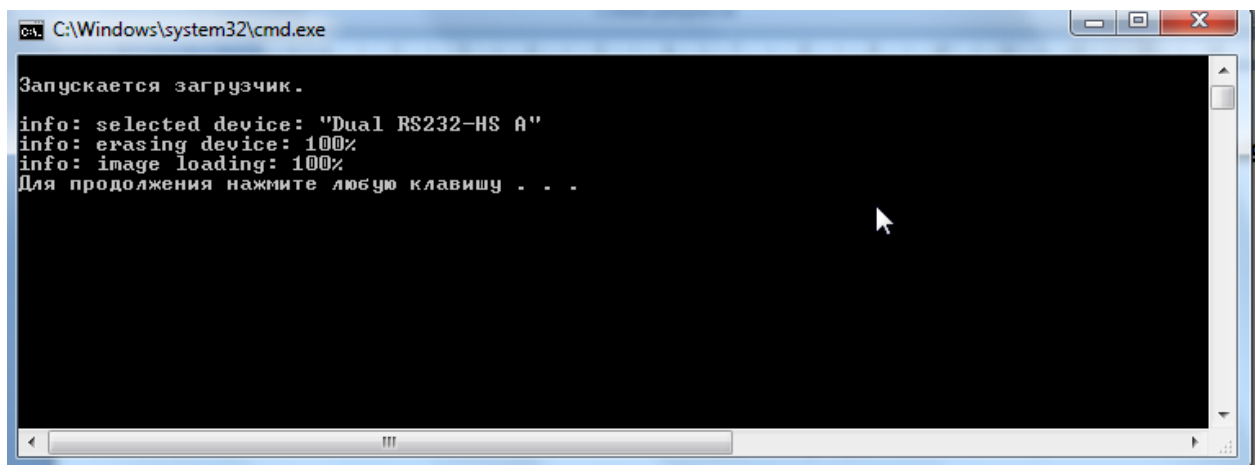


Рис. 10. Результат загрузки на плату

9) Результатом загрузки программы на плату должно стать мигание трёх светодиодов. Если этого не произошло сразу после загрузки программы, то нажмите кнопку SW5 для сброса процессора, светодиоды должны замигать.

Если светодиоды продолжают мигать - всё сделано верно! В случае возникновения вопросов обращайтесь в службу поддержки по адресу www.multiclet.com

Дополнительно

1. В папке Examples/LDM-МCp04 находится пример работы с UART, рассмотрим его подробнее:

```
1) #include <HDL51001_ccf.h>
2) #include <uart.h>
3)
4) void main()
5) {
6) UART_InitTypeDef UART_InitStructure;
7) GPIO->BPS = 0x00000F00;           //alternative port function for uart3
8)
9) UART_InitStructure.BaudRate = 38400; //set baudrate
10) UART_InitStructure.TypeParity = 0x00000000; //parity control type
11) UART_InitStructure.Parity = 0x00000000; //enable parity control
12) UART_InitStructure.FlowControl = 0x00000000; //enable cts/rts
13) UART_InitStructure.Mode = 0x00000003; //rx enable - 1 bit, tx enable - 2 bit (rx + tx en)
14)
15)     uart_init(UART3, &UART_InitStructure);
16)     // DM2UART(UART3, 0x00000000, 0x00000800);
17)
18) uart_send_str(" Hello, I'm MultiClet processor from Russia, Ekaterinburg!", UART3);
19) uart_send_str(" Привет! Я Первый Мультиклеточный процессор из
    Екатеринбург!", UART3);
20)
21) GPIO->DIR = ((uint32_t)0x00007000);
22) GPIO->OUT = ((uint32_t)0x00007000);
23)
24)     while(UART_FIFO_TX_EMPTY(UART3) == 0);
25)     //UART_SEND_BYTE(0xAB, UART3);
26) }
```

/*

В данном примере подключена обязательная библиотека HDL51001_ccf.h и библиотека интерфейса UART - uart.h в строках №1,2. Библиотека сторожевого таймера не подключена, поэтому будет происходить сброс процессора каждые 52 с и программа будет исполняться с начала.

- в строке №6 задаётся структура UART_InitStructure, имеющая тип UART_InitTypeDef;
- в строке №7 выводится, соответствующие UART3 (на данной плате UART3 подключён к микросхеме FTDI, которая создаёт USB-COM, см. рис.4) конфигурируются как альтернативные функции;
- в строке №9, задавая элемент структуры BaudRate, устанавливается скорость работы UART в 38400 бит/сек;

- в строке №10 устанавливается значение на чётность или нечётность, необходимо проверить данные (0 - чётность)
- в строке №11 устанавливается разрешение проверки на чётность (0 - запрещено)
- в строке №12 устанавливается разрешение аппаратного контроля данных (0 - запрещено)
- в строке №13 устанавливается разрешение работы UART на приём и передачу (0 – запрещено, 1 – только приём, 2 – только передача, 3 – приём и передача разрешены)
- в строке №15 происходит инициализация UART3 при помощи структуры UART_InitStructure (не заданные элементы структуры конфигурируются нулями по умолчанию)
- в строке №16 записан вызов функции DM2UART(), которая выдаёт содержимое памяти данных через UART3, начиная с 0-го адреса и размером 0x800 байт (2048 байт)
- в строках №18,19 записан вызов функции вывода строки посредством UART
- в строке №21,22 приведена конструкция зажигания светодиодов
- в строке №24 заложен контроль буфера передачи на пустоту
- в строке №25 записана посылка байта 0xAB через UART3

Примечание: для получения дополнительных сведений о регистрах интерфейса UART воспользуйтесь [руководством по эксплуатации процессора](#) MCp0411100101 (загрузить в разделе «Поддержка», «Техническая документация и ПО» на официальном сайте www.multiclet.com). Строки №16,20 закомментированы; для вывода памяти данных и байта 0xAB - раскомментируйте данные строки.

*/

2. Для получения данных посредством UART на ПК необходимо:

- подключить отладочную плату к компьютеру при помощи mini-USB кабеля
- скомпилировать проект и загрузить его на плату
- воспользоваться программой монитор СОМ-порта

3. Результатом будет приветствие от мультиклеточного процессора, см. рис 11.

#	Time	Received	ASCII
000001	17:35:10.065	20 48 65 6C 6C 6F 2C 20 49 27 6D 20 4D 75 6C 74 69 43 6C 65 74 20 70 72 6F 63 65 73 73 6F 72 20	Hello, I'm MultiClet processor
000002	17:35:10.065	66 72 6F 6D 20 52 75 73 73 69 61 2C 20 45 68 61 74 65 72 69 6E 62 75 72 67 21 20 CF F0 E8 E2 E5	from Russia, Ekaterinburg! Привет
000003	17:35:10.081	F2 21 20 DF 20 CF E5 F0 E2 FB E9 20 CC F3 EB FC F2 E8 EA EB E5 F2 EE F7 ED FB E9 20 EF F0 EE F6	т! Я Первый Мультиклеточный проц
000004	17:35:10.081	E5 F1 F1 EE F0 20 E8 E7 20 C5 EA E0 F2 E5 F0 E8 ED E1 F3 F0 E3 E0 21	ессор из Екатеринбурга!



Рис. 11. Результат работы программы



Параметры для настройки монитора СОМ-порта:

- скорость работы - 38400 бит/с
- размер одной посылки – 8 бит
- количество стоп битов – 1
- контроль четности отсутствует

Для удобства отображения информации при выводе памяти данных рекомендуется выравнивание по 32 байта в линии, сгруппированных по 8 байт.