

Для начала работы с отладочными платами LDM-HELPER-MB501-FULL-T2P8 и LDM-HELPER-MCP042RQ256-FULL необходимо наличие следующих

КОМПОНЕНТОВ:

- операционная система Windows (в данном примере рассматривается ОС Windows, также пользователи могут работать в среде Linux);
- отладочные комплекты LDM-HELPER-MB501-FULL-T2P8 и LDM-HELPER-MCP042RQ256-FULL;
- 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).
- Опционально: IDE Geany с плагином mc-dbg. Установку можно произвести вручную при помощи нашего инсталлятора или вручную согласно инструкции по программному обеспечению. Установочный комплект можно загрузить в разделе «[Техническая документация и ПО](#)» на официальном сайте www.multiclet.com.
- Программный монитор COM-порта.

Расположение используемых в данной инструкции разъемов и индикаторов:

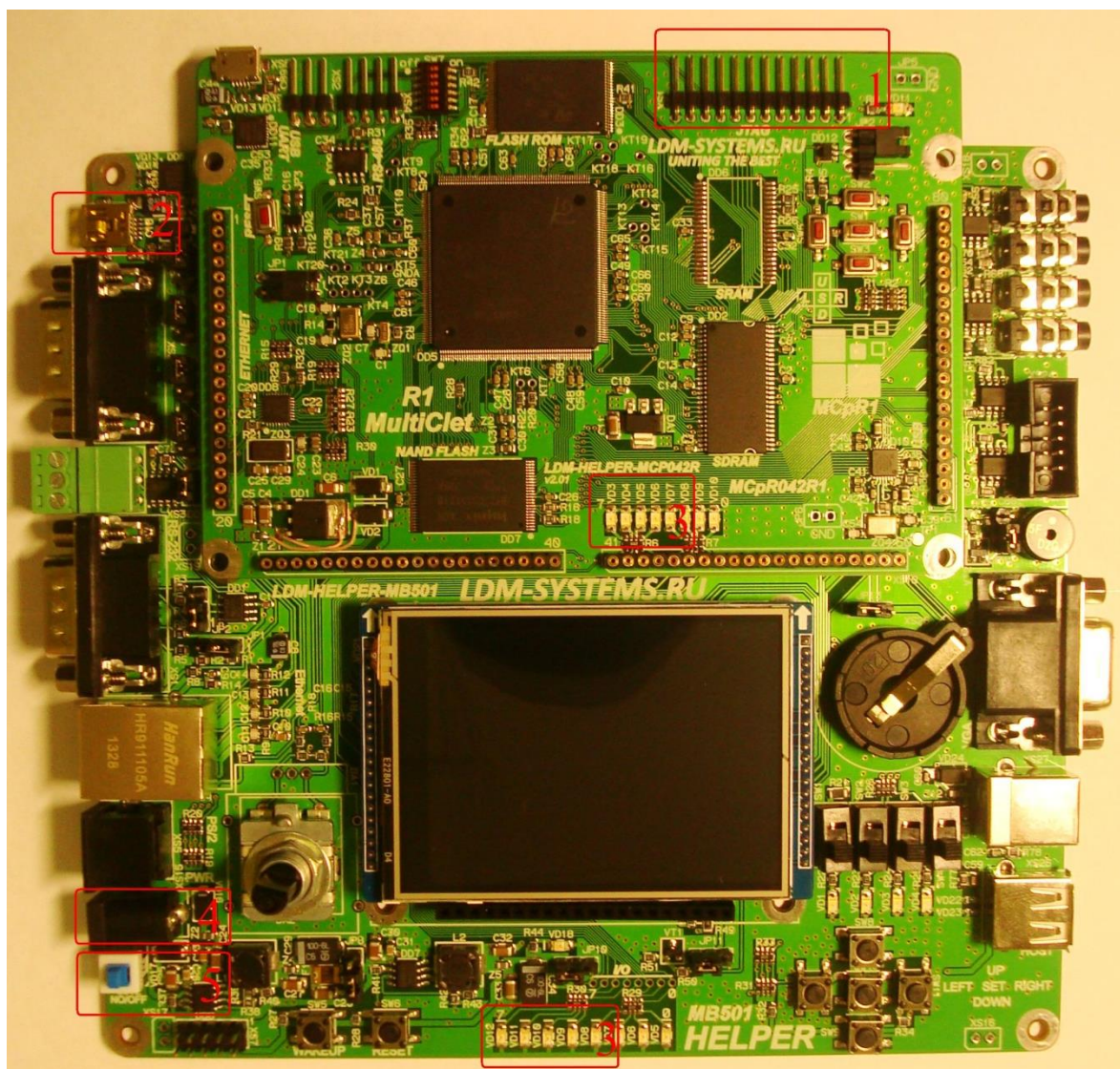


Рис 1. Расположение используемых в данной инструкции разъемов и индикаторов:

- 1) Разъем JTag.
- 2) Разъем Uart.
- 3) Светодиоды, подключенные к порту А.
- 4) Питание.
- 5) Кнопка включения.

Перед началом работы:

- 1) Установите инструментальный программный пакет.
- 2) Установите плату LDM-HELPER-MCP042RQ256-FULL на плату LDM-HELPER-MB501-FULL-T2P8.
- 3) Подключите питание к плате.
- 4) Подключите JTag к плате.

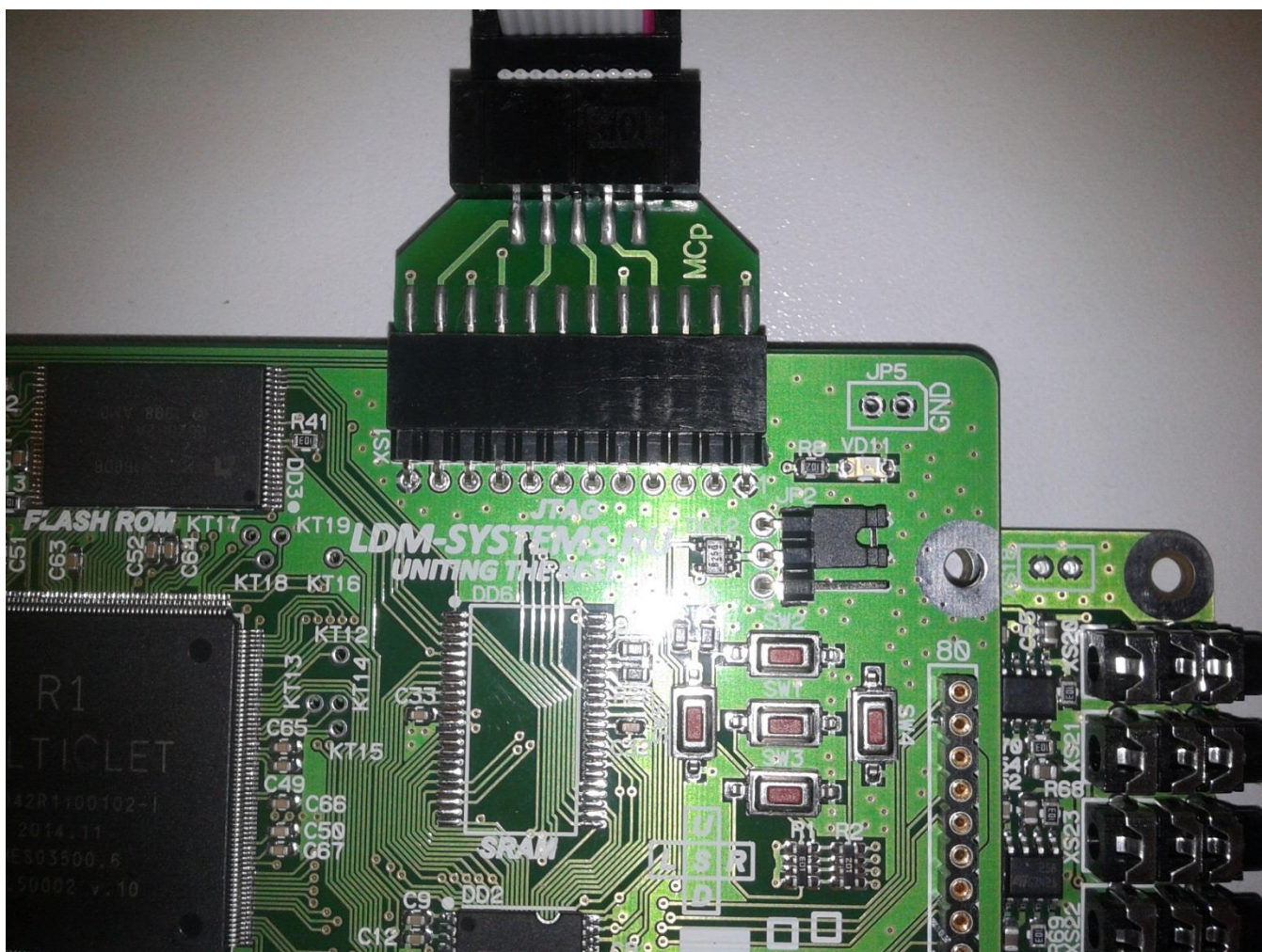


Рис 2. Верное подключение JTag к плате

- 5) Подключите mini-usb кабель к отладочной плате в разъем 2 (для работы с Uart).
- 6) Включите питание нажатием кнопки 5 (см. Рис. 1). На плате загорятся индикаторы VD10 и VD11, а также подсветка на дисплее.



Рис 3. Кабель mini-USB

- 4) Установите драйвер FTDI. Затем подключите с помощью кабеля mini-USB разъем USB вашего ПК к устройству JTag.
- 5) Опционально: установите IDE Geany с плагином mc-dbg. Установку можно произвести вручную при помощи нашего инсталлятора или вручную согласно инструкции по программному обеспечению.
- 6) Перезагрузите ваш ПК.

ПРИМЕР 1.

Программа на C, мигающая светодиодами

Программа расположена в примерах программ в папке *leds_c*. Код программы приведен ниже:

```
#define __IO volatile /*!< defines 'read / write' permissions */
//Описание структуры портов ввода-вывода
typedef struct
{
    __IO unsigned IN;
    __IO unsigned OUT;
    __IO unsigned DIR;
    __IO unsigned MSK;
    __IO unsigned POL;
    __IO unsigned EDG;
    __IO unsigned BPS;
} GPIO_TypeDef;
//Описание адреса порта A
#define GPIOA ((GPIO_TypeDef *) 0xC00F0000)

int main(void) {
    //Конфигурируем 26,28,29,30 пины порта A на вывод
    GPIOA->DIR= 0x3a000000;
    //Выставляем на 26,28,29,30 пинах порта A логическую 1
    GPIOA->OUT= 0x3a000000;
    while(1){
        unsigned i;
        //Цыкл задержки
        for(i=0;i<0xffff;i++);
        //Моргаем светодиодами
        GPIOA->OUT= GPIOA->OUT^0x3a000000;
    }
    return 0;
}
```

Компиляцию и загрузку программы можно выполнить при помощи командной строки или через IDE Geany с плагином mc-dbg.

Вариант 1. Компиляция и загрузка примера 1 через командную строку.

1) Соберите объектный файл leds.c из папки с примерами программ при помощи команды mc-icc. Пример команды:

```
mc-icc -lccdir=C:\bin -Wl-M -g -Wa—arch=MCP042R100102 -target=mcp -c
'C:\examples\leds_c\leds.c' -o 'C:\examples\leds_c\leds.o'
```

где C:\bin - путь до папки с файлом mc-icc, C:\examples\ - путь до папки с примерами программ.

2) Соберите бинарный файл при помощи mc-ld. Пример команды:

```
mc-ld 'C:\examples\crt0.o' 'C:\examples\leds_c\leds.o' -T C:\examples\script.txt --  
output='C:\examples\leds_c\leds.imagebin'
```

3) Загрузите бинарный файл на плату при помощи mc-jtagloader. Пример команды:

```
mc-jtagloader 'C:\examples\leds_c\leds.imagebin'
```

В процессе загрузки программы на плату будет производиться мигание светодиодами.

4) Перезагрузите плату при помощи нажатия кнопки RESET.

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

Вариант 2. Компиляция и загрузка примера 1 через IDE Geany.

В связи с тем, что текущая версия компилятора mc-icc не генерирует отладочную информацию, отладка программ, написанных на C, временно недоступна. Следите за выходом новых версий компилятора и отладчика на нашем сайте www.multiclet.com.

Прежде всего, при первом запуске IDE Geany, необходимо активировать плагин MC-DBG и настроить его.

Инструкция по настройке IDE Geany с плагином MC-DBG при первом запуске, созданию и настройке проектов:

- 1) Запустите Geany с помощью ярлыка на вашем рабочем столе
- 2) В главном меню выберите пункт *Инструменты/Менеджер модулей* и активируйте модуль *MC-DBG* (см. Рис 4). После успешной активации откройте вкладку MC-Files на боковой панели и MC-DBG на нижней панели.(см. Рис 5).

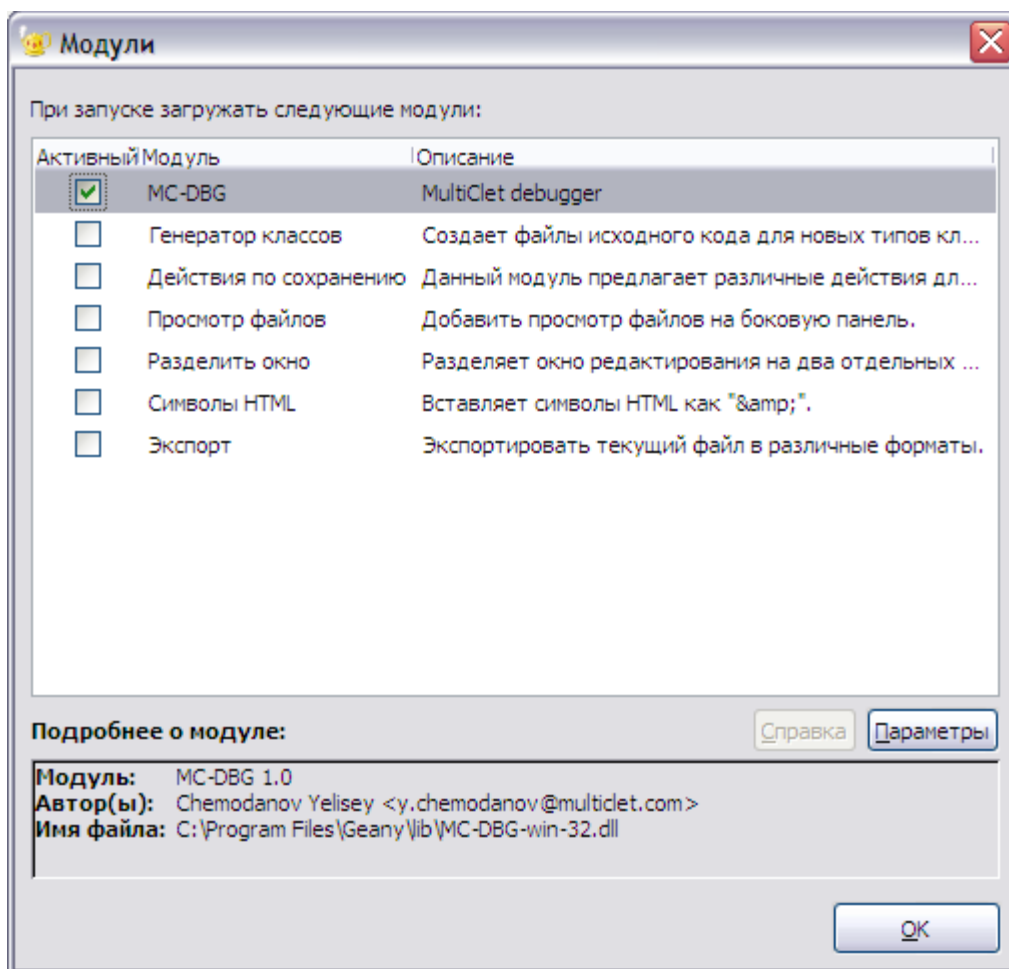


Рис 4. Окно Менеджера модулей с активированным плагином MC-DBG

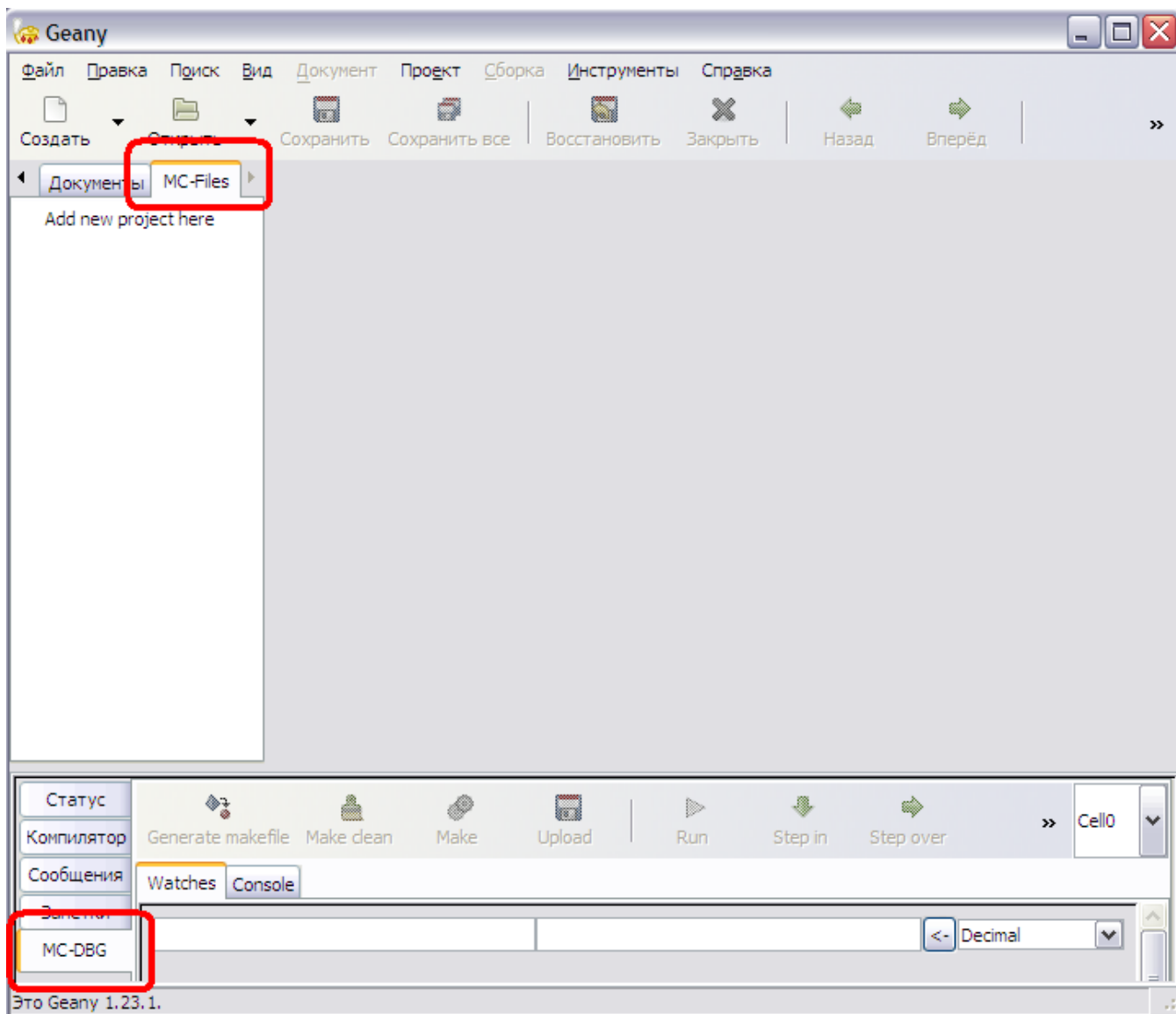


Рис 5. Вид окна Geany после успешной активации плагина MC-DBG

- 3) В контекстном меню элемента **Add new project here** выберите пункт **New**. Создайте новый проект. Если данная операция была выполнена успешно, его название отобразится в списке текущих проектов MC-Files вместо элемента **Add new project here**.
- 4) Нажмите правой кнопкой мыши на название вашего проекта для вызова контекстного меню. Выберите вкладку **Properties**. Во вкладке **Files** при помощи кнопки **Add** добавьте к проекту файлы **crt0.o** и **leds.c**. Включенные файлы в проект отобразятся в списке согласно порядку их линковки. При сборки проектов написанных на C, первым всегда должен линковаться файл **crt0.o** (кнопками Up и Down установите **crt0.o** на первое место в списке) (см Рис. 6).

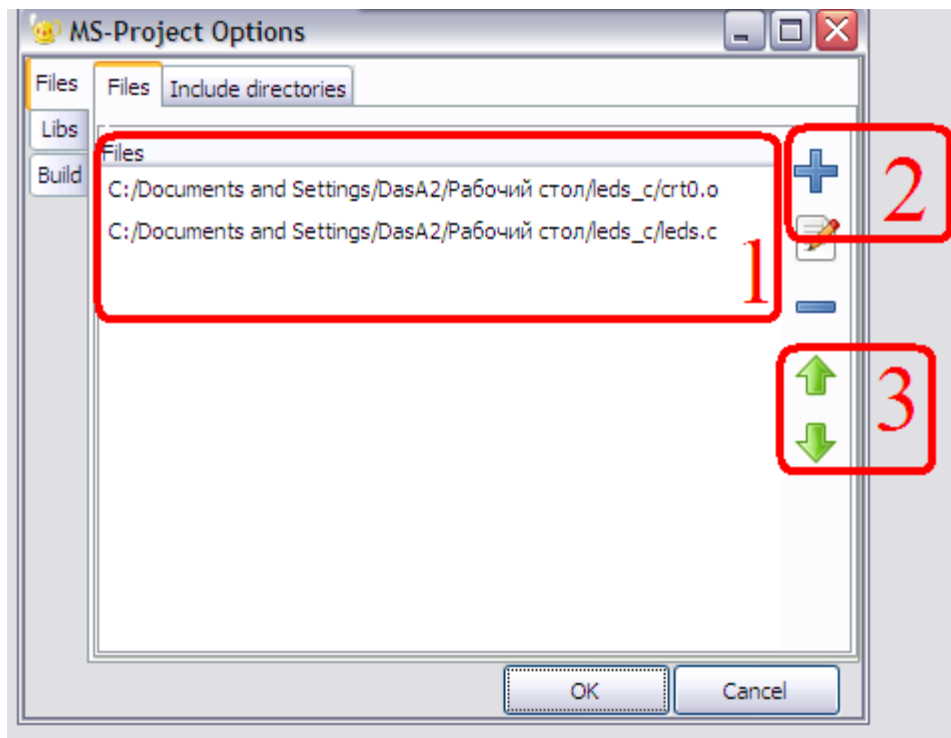


Рис 6. Вид окна *Geany Properties* после успешного добавления файлов из текущего примера.

1 – Список линкуемых файлов.

2 – Кнопка *Add*

3 – Кнопки *Up* и *Down*

- 5) Перейдите во вкладку **Build** и настройте опции для ассемблера, компилятора **C** и линкера (см Рис. 7, Рис. 8 и Рис. 9). Во вкладке **C Compiler** в опцию **-lccdir=** необходимо вписать путь до папки **MultiClet/SDK/bin** (например **-lccdir=C:/MultiClet/SDK/bin**). После успешной настройки примените их нажатием кнопки **OK**.

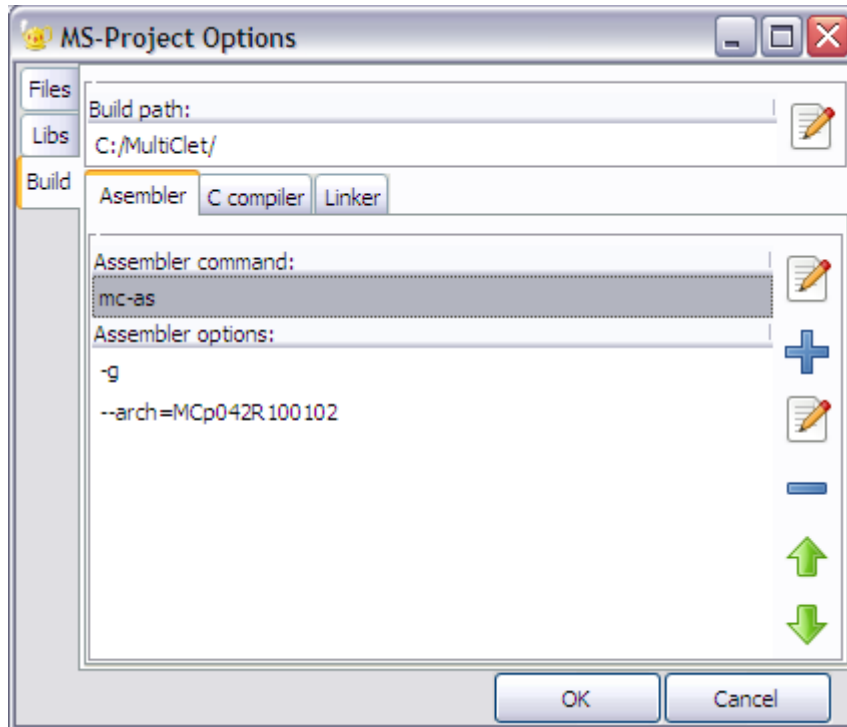


Рис 7. Пример настроек опций ассемблера.

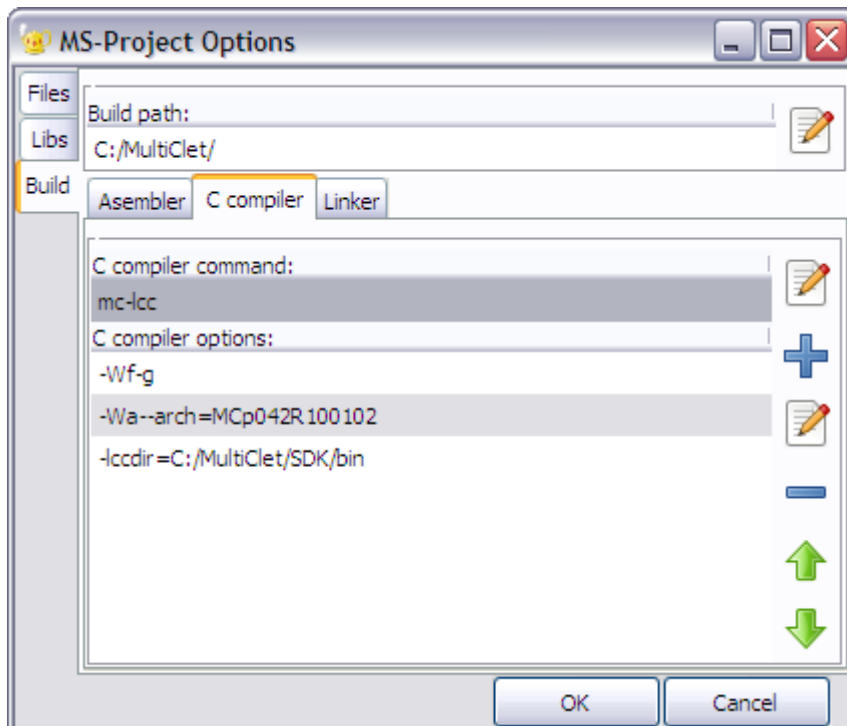


Рис 8. Пример настроек опций компилятора C.

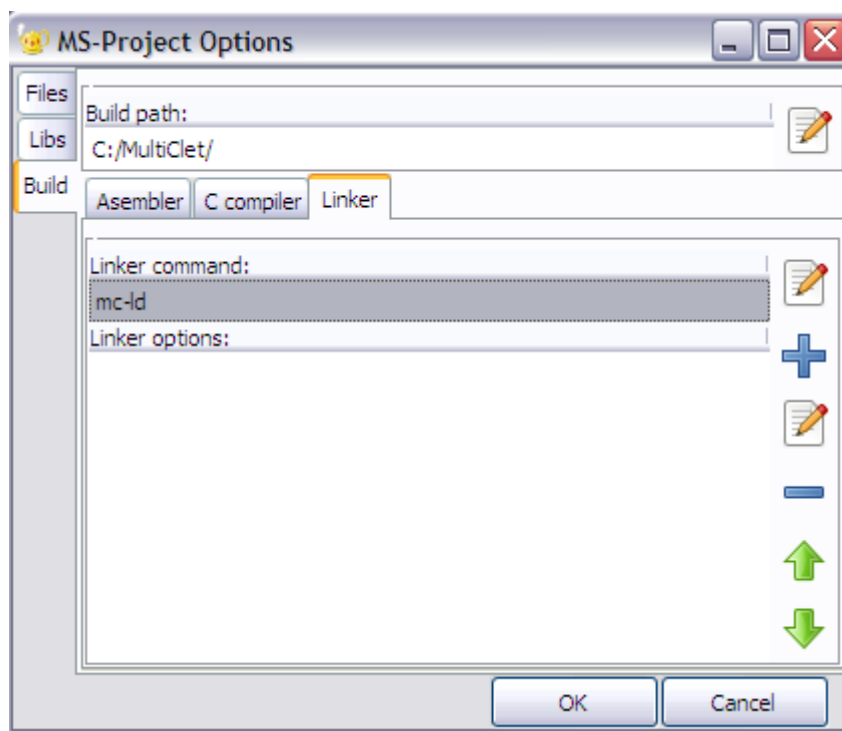


Рис 9. Пример настроек опций линкера.

- 6) Убедившись, что ПК подсоединен к плате по JTag, Выполняете компиляцию кнопкой Make и загрузку на плату кнопкой Upload. Кнопки располагаются на панели MC-DBG(см. Рис 10).

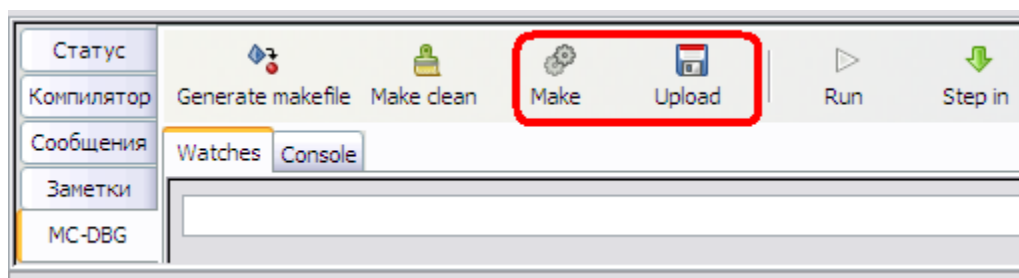


Рис 10. Расположение кнопок Make и Upload на панели MC-DBG.

- 7) После успешной загрузки, перезагрузите плату при помощи нажатия кнопки RESET.

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

ПРИМЕР 2.

Эхо сервер на ассемблере для Uart

Программа расположена в примерах программ в папке *UartExo_asm*. Код программы приведен ниже:

```
.alias UART0_CR      0xC0000108
.alias UART0_ST      0xC0000104
.alias UART0_BDR     0xC000010C
.alias UART0_DATA    0xC0000100

.alias GPIOC_DATA    0xC01F0200 ;no use in the future
.alias GPIOC_IN      0xC00F0200
.alias GPIOC_OUT     0xC00F0204
.alias GPIOC_DIR     0xC00F0208
.alias GPIOC_MSK     0xC00F020C
.alias GPIOC_POL     0xC00F0210
.alias GPIOC_EDG     0xC00F0214
.alias GPIOC_BPS     0xC00F0218
.alias Get_byte 2

.text

start:
    jmp initUART0
    getl 0xC00
    wrl @1, GPIOC_BPS
    getl 0x1A
    wrl @1, UART0_BDR
complete

;;;; Configurate Uart ;;;;
initUART0:
    jmp MainLoop
    getl 0x0000000A
    wrl @1, UART0_DATA
    getl 0x00000003
    wrl @1, UART0_CR
complete

;RXD handler
Rxd_int:
    jmp buf_TXD
    rd1 UART0_DATA
    setl #Get_byte, @1
complete

;test FIFO on full
buf_TXD:
    rd1 UART0_ST
    getl 0x00000200
    and @1, @2
    jne @1, buf_TXD
    je @2, Send_byte
complete
```

```
;Send inverse byte
Send_byte:
    jmp MainLoop
    getl #Get_byte
    wrl @1, UART0_DATA
complete

;;Loop
MainLoop:
    rd1 UART0_ST
    getl 0x00000001
    and @2,@1
    je @1, MainLoop
    jne @2, Rxd_int
complete
```

В параграфах *start* и *initUART0* происходит инициализация Uart. В параграфе *Rxd_int* происходит чтение данных с Uart. В параграфе *buf_TXD* производится контроль на переполнение выходного сдвигового регистра. В параграфе *Send_byte* прочитанный байт возвращается по Uart. В параграфе *MainLoop* производится контроль статусного регистра Uart на предмет поступления новых данных.

Компиляция и загрузка примера 1 через командную строку.

Производится по аналогии с 1 примером за исключением первой команды (вместо неё вызываем *mc-as*). Примеры команд:

```
mc-as      —arch=MCp042R100102  C:\UartExo_asm\exo.asm
mc-ld      C:\UartExo_asm\exo.o
mc-jtagloader  C:\UartExo_asm\image.bin
```

Компиляция и загрузка примера 1 через IDE Geany.

Производится согласно инструкции по настройке IDE Geany с плагином MC-DBG при первом запуске, созданию и настройке проектов, приведенной выше. В проект следует включить только файл *exo.asm*.

Примечание: для получения дополнительных сведений о регистрах интерфейса UART воспользуйтесь руководством по эксплуатации процессора R1 (загрузить в разделе «Поддержка», «Техническая документация и ПО» на официальном сайте www.multiclet.com).

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 COM-порта

Параметры для настройки монитора COM-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.

Отладочный комплект LDM-HELPER-MCP042RQ256 предназначен для ознакомления с реализацией мультиклеточной архитектуры и тестирования реконфигурации клеток.

Рекомендации по эксплуатации отладочного комплекта с микропроцессором MULTICLET R1-1.

Мультиклеточный процессор **MULTICLET R1**, в соответствии с классификацией Леонида Черняка (см. [статью](#)), относится к динамически реконфигурируемым, однако это процессор на кристалле и, вследствие независимости машинного кода, перераспределение ресурсов (клеток) в отличие от существующих аналогов на FPGA происходит без остановки или перезагрузки процессора и без потери информации. Таким образом, **MULTICLET R1** представляет собой новый класс динамически реконфигурируемых процессоров. В настоящее время в мире не существует аналогов микропроцессоров на кристалле с реализованной функцией динамической реконфигурации (Подробнее читайте в [статье](#)). Примеры программ для освоения данной функции приведены в разделе [«Техническая документация и ПО»](#) на официальном сайте www.multiclet.com.

- Плата не предназначена для работы с внешним ПЗУ (эта возможность будет предоставляться в комплекте с процессором **MULTICLET R1**). На плате PROM не доступна, в связи с чем, работа DMA с PROM невозможна. Работа с PROM возможна при ее корректном подключении к процессору **MULTICLET R1-1** (см. [Руководство по эксплуатации для MULTICLET R1](#)).
- Использование прерываний находится в состоянии тестирования, рекомендуем смотреть изменение в документации.
- При контроле чтения и записи — все параграфы должны иметь каноническую форму (чтение, модификация, запись).
- Рекомендуется использовать Uart на LDM-HELPER-MB501-FULL-T2P8, помеченный на Рис. 1, как разъем 2.

Для продолжения ознакомления с процессором **MULTICLET R1** мы рекомендуем вам изучить [Руководство пользователя по программному обеспечению](#) для **MULTICLET R1** и [Руководство по эксплуатации для MULTICLET R1](#).