



Отладочная плата
LDM-МСр0411100101-Q208 Evolution



Отладочная плата

LDM-МСр0411100101-Q208 Evolution



СДЕЛАНО В РОССИИ

СОДЕРЖАНИЕ

	Стр.
ВВЕДЕНИЕ	3
1 ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ	4
2 ОПИСАНИЕ И РАБОТА	5
2.1 Назначение изделия	5
2.2 Описание микропроцессора МСр0411100101-Q208I	6
2.3 Технические характеристики микропроцессора	6
2.4 Техническое описание	8
2.5 Устройство и работа	11
2.5.1 Органы управления	11
2.5.2 Установка драйверов и ПО	14
2.5.3 Использование устройства	19
2.6 Принципиальные электрические схемы отладочной платы	32
2.7 Монтажные чертежи	38
2.8 Трассировка по слоям	39
3 ЭКСПЛУАТАЦИЯ, ХРАНЕНИЕ И ТРАНСПОРТИРОВАНИЕ	41

ВВЕДЕНИЕ

Отладочная плата LDM-MCp0411100101-Q208 Evolution предназначена для разработки высокопроизводительных энергоэффективных электронных устройств на базе мультиклеточного процессора MCp0411100101-Q208I Российской компании ЗАО «Мультиклет». Сбалансированная архитектура платы упрощает процесс тестирования и отладки программного обеспечения, позволяет минимизировать финансовые затраты и сократить сроки проектных работ. Средство разработки будет интересно профессионалам и начинающим проектировщикам.

В настоящем руководстве по эксплуатации описаны характеристики и правила работы с платой LDM-MCp0411100101-Q208 Evolution (рис. 1). Приведен порядок установки ПО и драйверов. В документе кратко изложены особенности и возможности микропроцессора MCp0411100101. Описана его внутренняя организация и описана широкая номенклатура периферийных устройств.

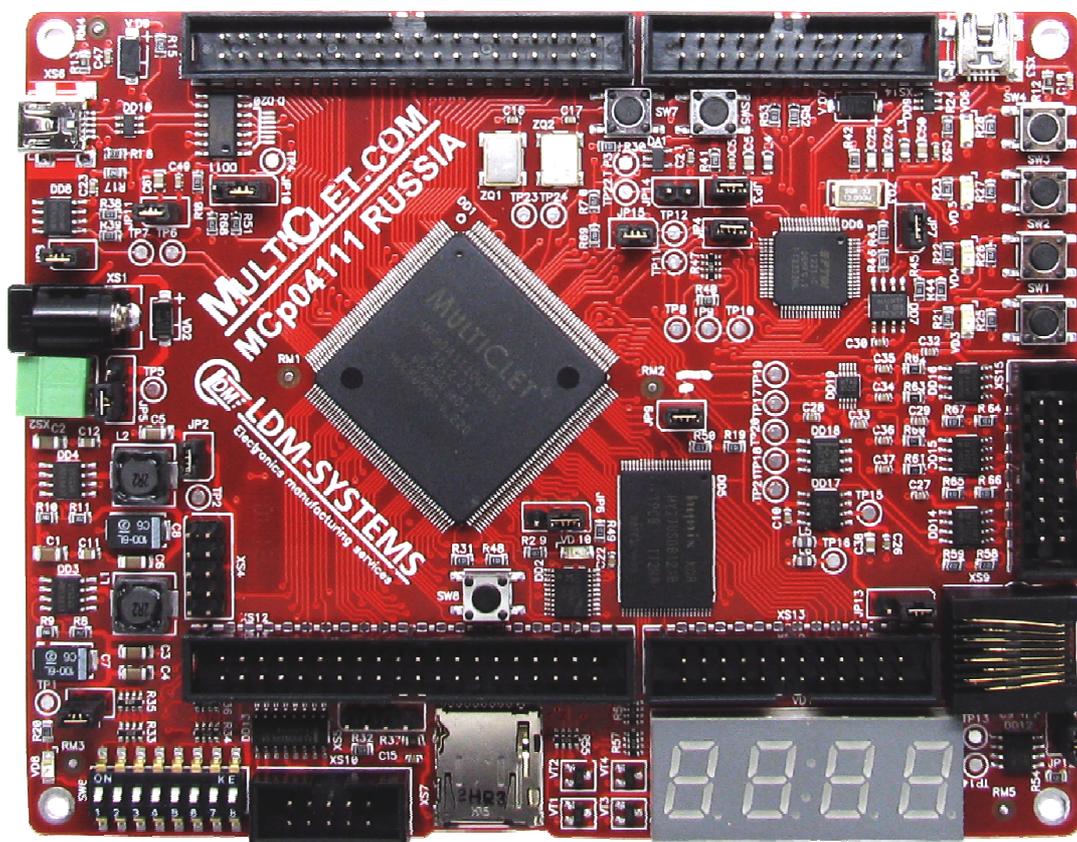


Рисунок 1. Общий вид отладочной платы LDM-MCp0411100101-Q208 Evolution

1 ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

МП – микропроцессор;

ПЗУ – постоянное запоминающее устройство;

ПП – память программ;

ПО – программное обеспечение;

ПД – память данных.

2 ОПИСАНИЕ И РАБОТА

2.1 Назначение изделия

Отладочная плата LDM-МСр0411100101-Q208 Evolution предназначена для разработки, отладки систем на базе мультиклеточного процессора МСр0411100101 производства ОАО «Мультиклет».

Отладочная плата LDM-МСр0411100101-Q208 Evolution позволяет разработчику максимально быстро освоить принципы работы с мультиклеточным процессором МСр0411100101-Q208I и управление его периферийными устройствами.

2.2 Описание микропроцессора МСр0411100101-Q208I

Мультиклеточный микропроцессор МСр0411100101-Q208I предназначен для решения широкого круга задач управления и цифровой обработки сигналов в приложениях, требующих минимального энергопотребления и высокой производительности, таких как:

- системы промавтоматики от интеллектуальных датчиков до систем управления;
- устройства управления двигателями;
- универсальные навигационные приемники ГЛОНАСС/GPS/Galileo/COMPASS(Китай)/IRNSS(Индия)/QZSS(Япония);
- мобильные телефоны;
- видеотехника 3D;
- автомобильная электроника для «интеллектуальных» бортовых систем, контролирующих дорожную обстановку и предупреждающих водителей об опасностях и пробках;
- системы безопасности, автоматически распознающих «свой / чужой».

2.3 Технические характеристики микропроцессора

Микропроцессор МСр0411100101-Q208I имеет в своем составе мультиклеточное процессорное ядро – первое процессорное ядро с принципиально новой пост-неймановской мультиклеточной архитектурой российской разработки. Мультиклеточный процессор состоит из 4 клеток (когерентных процессорных блоков), объединенных интеллектуальной коммутационной средой.

Особенности:

- число клеток – 4;
- разрядность процессора - 32/64 бита;
- ПД- 128 Кб (4*4К*64);
- ПП – 128 Кб (4*4К*64);

- ПЗУ - в исполнении «1» отсутствует, для хранения исполняемого кода должно применяться внешнее последовательное FLASH ПЗУ (на отладочной плате установлена FLASH XCF04S XILINX);
- блок операций над числами с плавающей запятой (в каждой клетке);
- тактовая частота — 100 МГц;
- производительность процессора — 2,4 Gflops.

Общие характеристики:

- корпус – QFP-208;
- условия эксплуатации – (-60...+125) °С;
- максимальная потребляемая мощность процессора: 1,08 Вт;
- напряжение питания (раздельное):
 - ядра – 1,8 В;
 - периферии – 3,3 В.

Периферийные устройства:

- 3 интерфейса SPI с селектором «ведомых» устройств (в режиме «ведущий»);
- 4 универсальных асинхронных приёмопередатчика UART с FIFO на прием/передачу;
- 2 интерфейса I2C (один «master» и один «slave»);
- интерфейс I2S;
- Ethernet контроллер 10/100 Мб/с;
- USB 1.1 FS (device) контроллер с последовательным внешним интерфейсом для подключения приемо-передатчика;
- 7 таймеров общего назначения;
- 4 порта ввода-вывода, общее количество вводов-выводов – 104;
- 4-х канальный контроллер ШИМ;
- сторожевой таймер.

2.4 Техническое описание

На рисунке 2 приведены основные блоки внутренней и внешней обвязки отладочной платы LDM-MCp0411100101-Q208 Evolution.

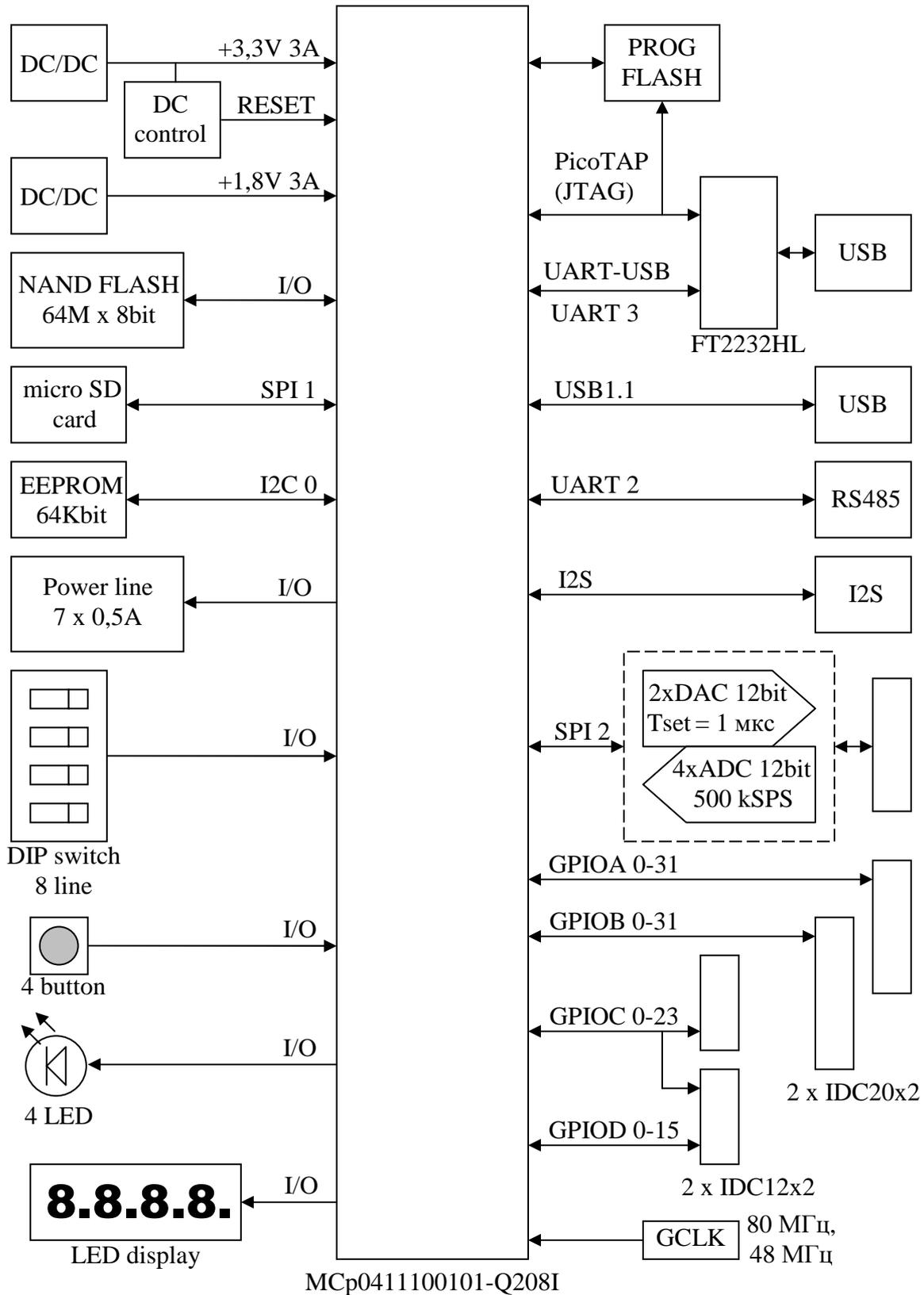


Рисунок 2. Блок-схема отладочной платы LDM-MCp0411100101-Q208 Evolution

Компоненты:

- процессор MСр0411100101-Q208I;
- тактовый генератор 80 МГц (ядро MСр0411100101);
- тактовый генератор 48 МГц (USB интерфейс);
- ЦАП – 2 канала, 12 бит, $T_{set} = 1$ мкс;
- АЦП – 4 канала, 12 бит, 500 кс/с;
- ОУ на выходе ЦАП и входе АЦП;
- I2C EEPROM 64 кб;
- NAND FLASH 64 МБ;
- FLASH программ (XCF04SVOG20C).

Интерфейсы:

- USB 1.1 FS, разъем Mini USB-A;
- UART-USB, разъем Mini USB-A (совместная работа с PicoTAP);
- I2S, разъем PLS-4;
- разъем micro SD карты;
- 7 силовых линий 0,5 А, 30 В;
- UART-RS485, разъем TJ4-8P8C.

Средства отладки:

- встроенный JTAG программатор PicoTAP;
- все порты МП выведены на 4 разъема;
- контрольные точки на плате.

Управление и индикация:

- кнопки: «reset», «wake-up», «nmi» и 4 для произвольного использования;
- светодиоды: «MP-ready», «power» и 4 для произвольного использования;
- четырехзначный семисегментный LED индикатор.

Электропитание:

- на плате установлены импульсные источники опорного напряжения 3,3 В, 3 А и 1,8 В, 3 А;
- при питании от порта USB с «Host» должно подаваться постоянное напряжение – 5 В, необходимо помнить об ограничении по току при питании от порта USB;
- максимальное потребление по цепи 5 В (ММП с максимальной производительностью на частоте 80 МГц, без коммуникаций – 400 мА);
- при подаче питания на коаксиальный разъем или клеммную колодку должно быть постоянное напряжение – 5-12 В;
- все линии питания имеют защиту от противовключения напряжения.

Физические характеристики:

- материал платы: FR-4, 1.5 мм, 4 слоя. С шелкографией и маской;
- габариты: 140 x 110 x 15 мм.

Комплект поставки:

- отладочная плата LDM-МСр0411100101-Q208 Evolution с МП МСр0411100101-Q208};
- USB-A - mini USB-B кабель;
- CD-R диск с ПО и технической документацией;
- фирменная упаковка.

2.5 Устройство и работа

2.5.1 Органы управления

Размещение компонентов на плате

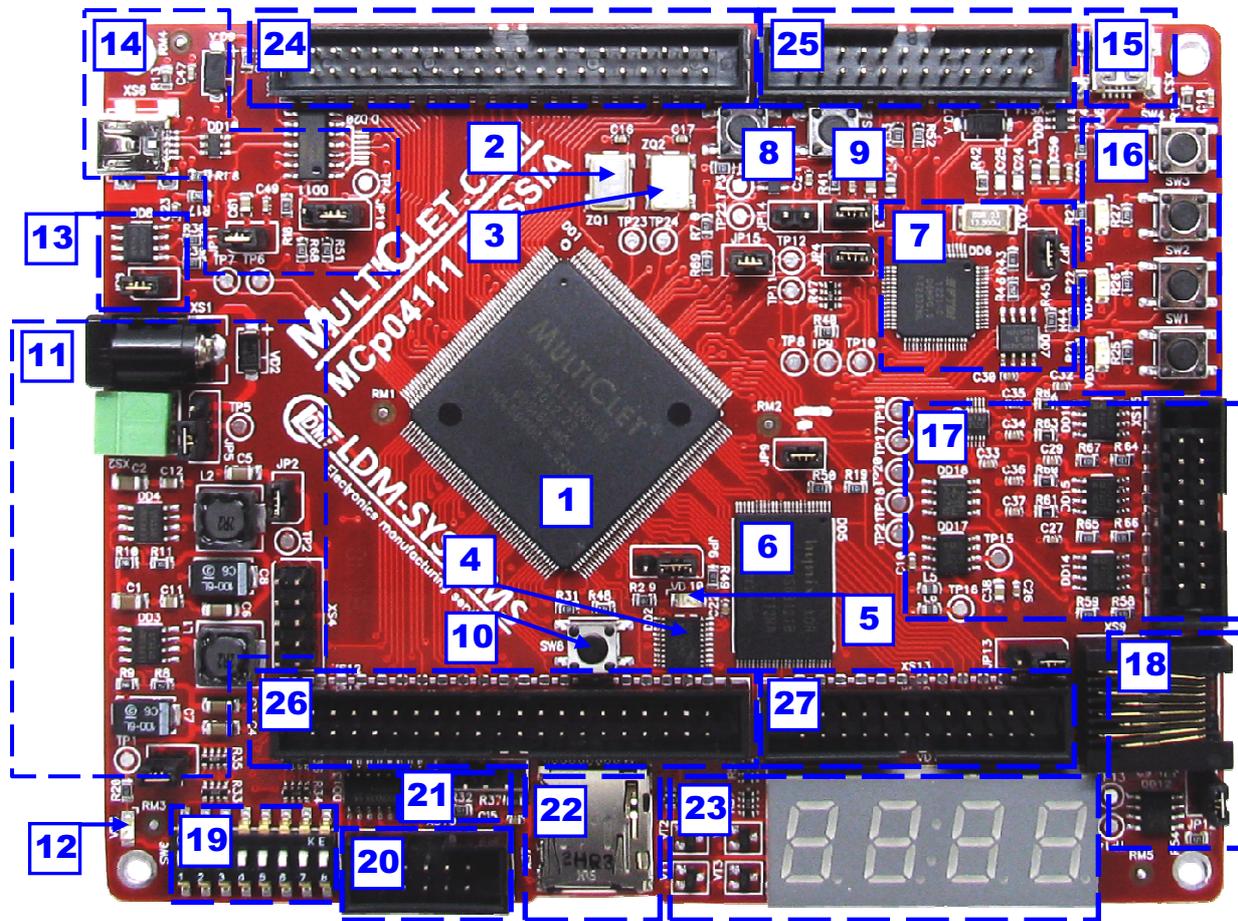


Рисунок 3. Размещение компонентов на плате LDM-MCp0411100101-Q208 Evolution

№	Назначение	№	Назначение
1	Процессор MCr0411100101-Q208I	15	Разъем miniUSB, PicoTAP+USB-COM
2	Тактовый генератор 80 МГц	16	Блок кнопок и светодиодов
3	Тактовый генератор 48 МГц	17	Блок АЦП, ЦАП
4	FLASH программ (XCF04SVOG20C)	18	Интерфейс RS485
5	Индикатор готовности процессора	19	Переключатели на 8 I/O
6	NAND FLASH 64 МБ	20	Силовые линии 7x0,5 А, 30 В
7	Блок PicoTAP+USB-COM	21	Протокол I2S, разъем PLS-4
8	Кнопка «wake-up»	22	Разъем micro SD карты
9	Кнопка «reset»	23	Индикатор LED 4 разряда
10	Кнопка «nmi»	24	Порт A (32 I/O)
11	Блок DC/DC преобразования	25	Порт D (16 I/O) + порт C (4 I/O)
12	Индикатор наличия питания	26	Порт B (32 I/O)
13	Память EEPROM 64 кб, протокол I2C	27	Порт C (20 I/O)
14	Интерфейс USB 1.1, разъем miniUSB		

Описание органов управления

Перемычки:

Номер	Позиция 1-2	Позиция 2-3	Штатная позиция
JP1	Подключение линии питания 3.3 В (питание линий I/O портов процессора)	-	1-2
JP2	Подключение линии питания 1.8 В (питание ядра процессора)	-	1-2
JP3	Подключение линии TCK PicoTAP к линии TCK процессора и PROM	-	1-2
JP4	Подключение линии RST PicoTAP к линии JNTRST процессора	-	1-2
JP5	Питание платы от внешнего источника (разъемы XS1, XS2)	Питание платы от USB порта (разъемы XS3, XS6)	1-2
JP6	Подключение линии TMS PicoTAP к линии TMS PROM (DD2) XCF04SVO20C	Подключение линии TMS PicoTAP к линии TMS процессора	1-2
JP7	Режим программатора PicoTAP	Вынут - режим USB-COM	1-2
JP8	Подключение линии SCL EEPROM к линии SCL протокола I2C процессора	-	1-2
JP9	Подключение линии PC16 процессора к линии активации памяти NAND FLASH (если память не используется, то джампер вынимается)	-	1-2
JP10	Режим настройки USB моста USB1T11 (MODE = 0)	Режим настройки USB моста USB1T11 (MODE = 1)	1-2
JP11	Подключение линии OE' USB моста USB1T11 к управляющей линии процессора	-	1-2
JP12	Подключение шунта 120 Ом на линию передачи интерфейса RS485	-	1-2
JP13	Подключение линии питания 3,3 В к выводу 1, 2 разъема XS9 интерфейса RS485 (если нет необходимости запитывать внешнее устройство по линии RS485, то джампер вынуть)	Подключение линии питания от разъемов XS1, XS2 к выводу 1, 2 разъема XS9 интерфейса RS485 (если нет необходимости запитывать внешнее устройство по линии RS485, то джампер вынуть)	1-2
JP14	Подключение линии процессора PC19 к выводу WDI контроллера уровня напряжения 3,3 В	-	Вынут
JP15	Подключение линии RST PicoTAP к линии глобального сброса RESET	-	1-2

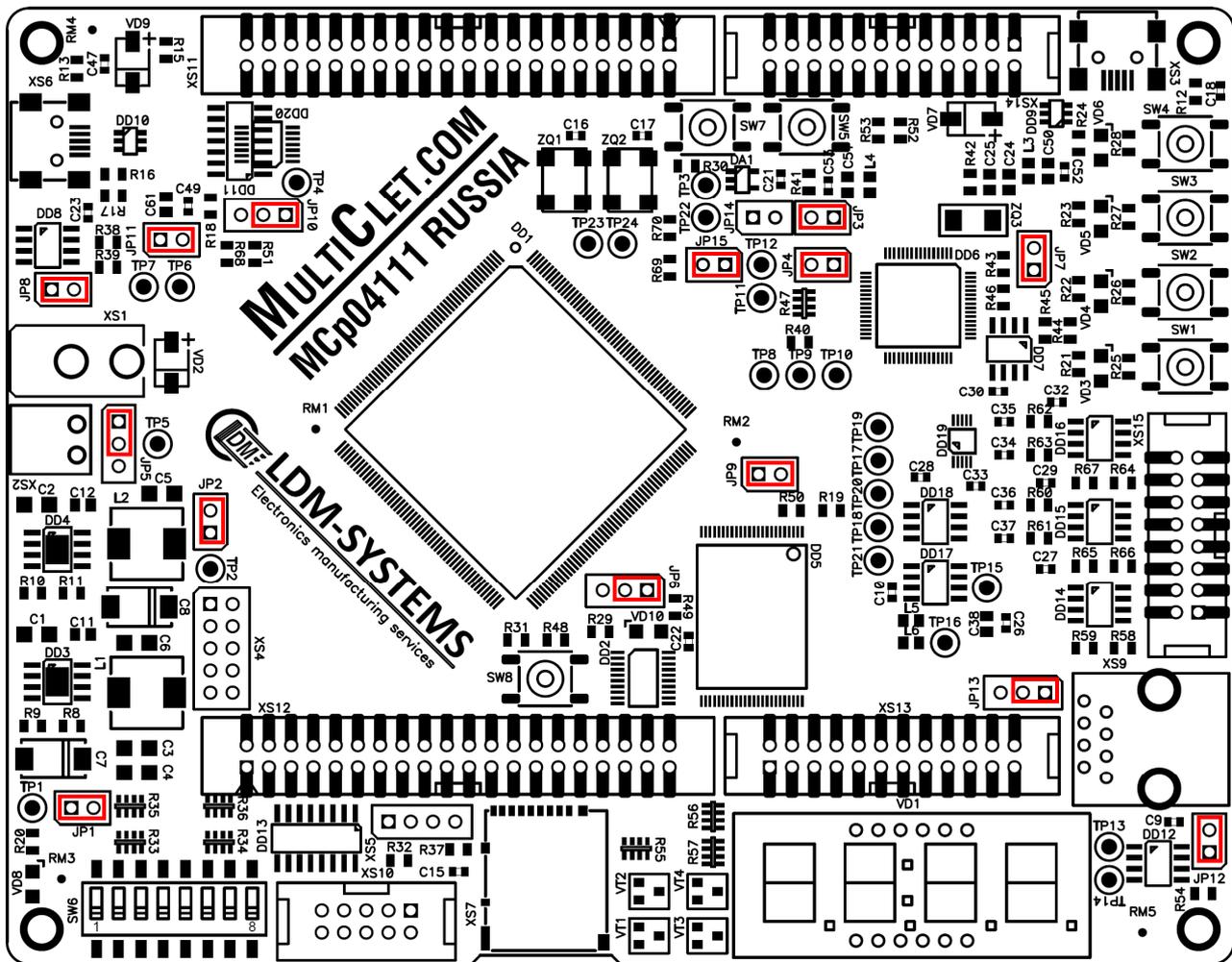


Рисунок 4. Штатные позиции переключателей на плате LDM-MCp0411100101-Q208 Evolution

Кнопки:

- SW1 – SW4 – пользовательские кнопки;
- SW5 – кнопка глобального сброса RESET;
- SW7 – выход из спящего режима «wake-up»;
- SW8 – немаскируемое внешнее прерывание «nmi».

Индикация:

- VD3 – VD6 – пользовательские светодиоды;
- VD8 – наличие питания +3,3 В;
- VD10 – память MCp готова, MCp в рабочем режиме «MP-ready».

2.5.2 Установка драйверов и ПО

Перед началом работы с платой необходимо установить драйвер микросхемы FT2232 интерфейса USB-COM и драйвер JTAG-контроллера PicoTAP. Необходимые драйверы содержатся на диске, входящем в комплект поставки, также их можно скачать с сайта производителей (www.goepel.com, www.ftdichip.com).

ШАГ 1: Установка драйвера микросхемы FT2232 интерфейса USB-COM

Linux

Последняя версия драйвера для Linux и инструкции по его установке находятся на сайте, упомянутом выше. Также вы можете воспользоваться драйвером, входящим в комплект поставки.

На оптическом диске из отладочного комплекта найдите файл CD:\DRV\FT2232\Linux\libftd2xx1.1.0.tar.gz:

1. Распакуйте файл в подходящую директорию на диске:

- `gunzip libftd2xx1.1.0.tar.gz;`
- `tar -xvf libftd2xx1.1.0.tar.`

2. Смените директорию на требуемую для архитектуры МП вашего ПК: `build/i386` - для 32-битной системы или `build/x86_64` - для 64-битной системы.

3. Под пользователем «root» скопируйте упомянутые ниже файлы в директорию `/usr/local/lib`:

- `cp libftd2xx.so.1.1.0 /usr/local/lib.`

4. Смените директорию на `/usr/local/lib`:

- `cd /usr/local/lib.`

5. Создайте ссылку на файлы:

- `ln -s libftd2xx.so.1.1.0 libftd2xx.so.`

6. Смените директорию на `/usr/lib`:

- `cd /usr/lib.`

7. Создайте ссылку на драйвер:

- In -s /usr/local/lib/libftd2xx.so.1.1.0 libftd2xx.so.

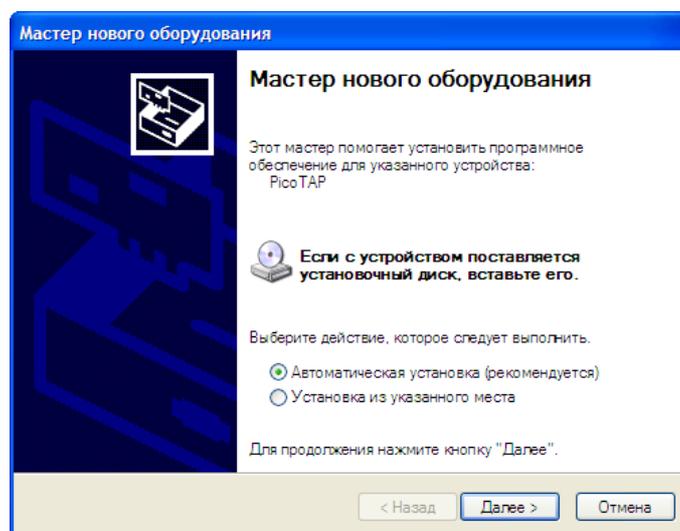
Windows

1. На входящем в комплект поставки диске найдите и запустите файл CD:\DRV\FT2232\Windows\CDM20824_Setup.exe, следуйте инструкциям установщика.

ШАГ 2: Установка драйвера JTAG-контроллера PicoTAP

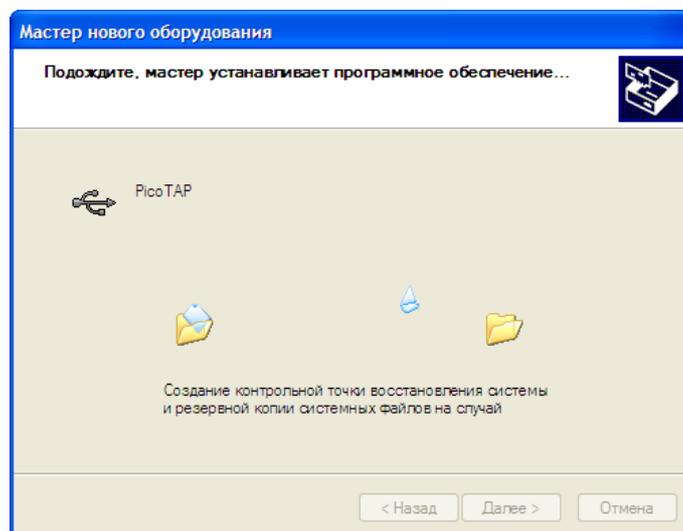
1. На входящем в комплект поставки диске найдите и запустите файл CD:\DRV\PicoTAP\Pico_TAP_ohneJtag(DRV).exe, следуйте инструкциям установщика.

2. Подключите плату LDM-MCp0411100101-Q208 Evolution к ПК (п. 2.5.3), появится окно оповещения:

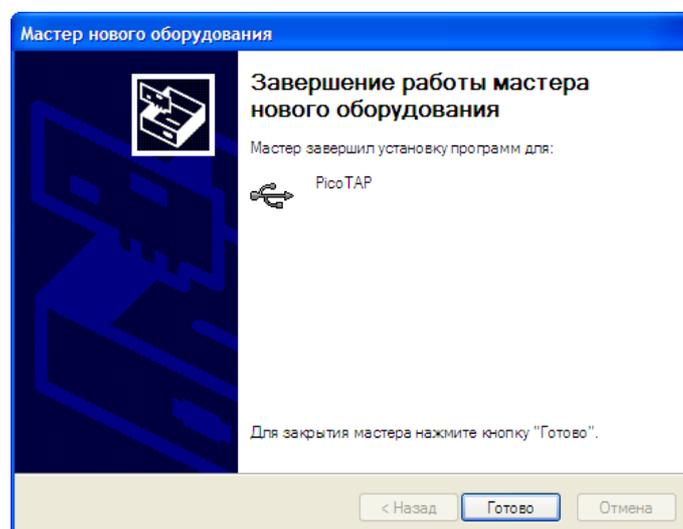


3. Выберите автоматическую установку ПО (Install the software automatically) и нажмите кнопку «Далее» (Next).

4. Если ОС Windows запросит подтвердить установку устройства PicoTAP не тестированное на совместимость с ней, то дайте подтверждение, нажмите кнопку «Продолжить» (Continue Anyway).



5. После окончания установки драйверов нажмите кнопку (Finish) в финальном окне.



6. Снова система оповестит о найденном устройстве PicoTAP. Это нормально, т.к. мост FT2232 состоит из двух портов PicoTAP. Повторите действия, начиная с п. 3.

7. После завершения установки, система оповестит, что устройство подключено и работает нормально.

ШАГ 3: Установка компилятора

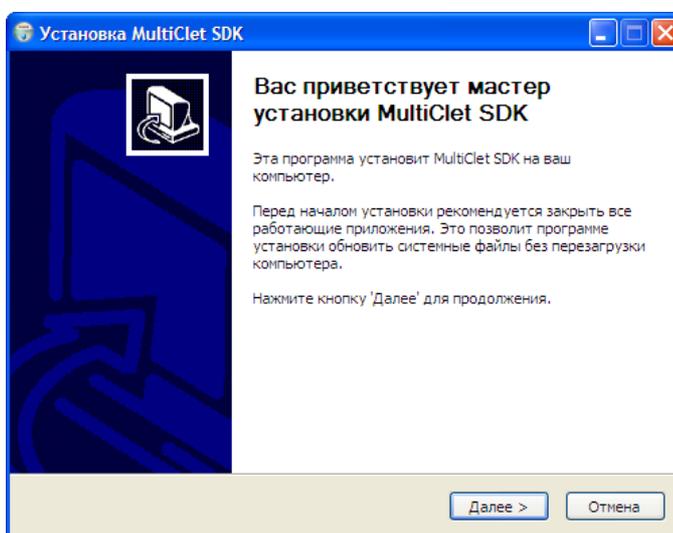
Дистрибутив компилятора содержится на диске, входящем в комплект поставки, также его можно скачать с сайта производителей (www.multiclet.com).

1. На входящем в комплект поставки диске найдите файлы и установите их:

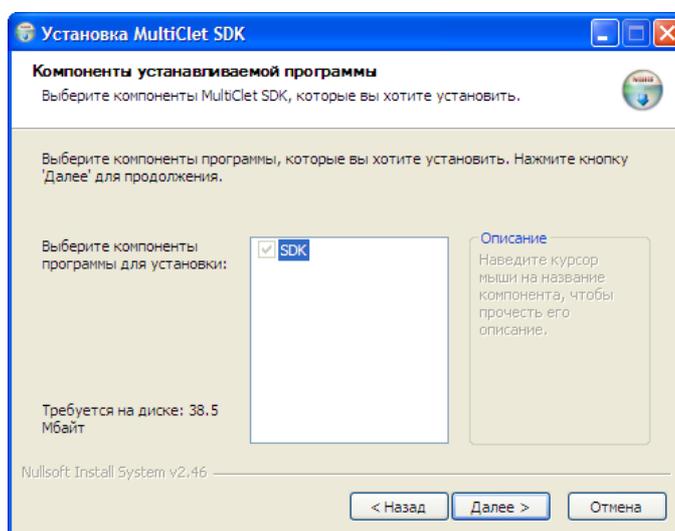
- Linux: CD:\Компилятор\Linux\MultiCletSDK.20121205.tar.gz;
- Windows: CD:\Компилятор\Windows\MultiCletSDK.20121205.exe.

На примере ОС Windows:

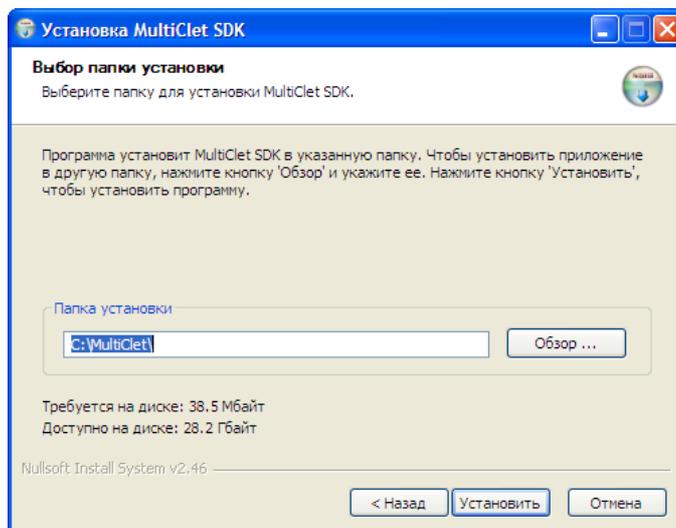
2. В окне приветствия нажмите кнопку «Далее»



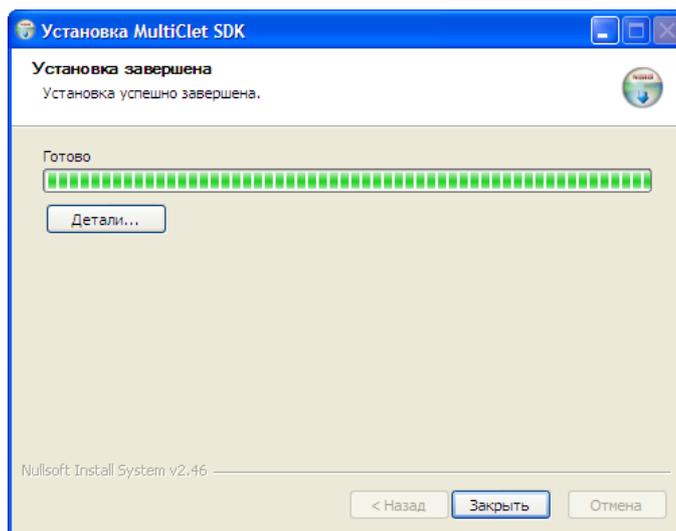
3. В окне «Компоненты устанавливаемой программы» нажмите кнопку «Далее»



4. В окне «Выбор папки установки» оставьте директорию без изменений и нажмите кнопку «Установить»



5. В окне «Установка завершена» нажмите кнопку «Закреть»



6. Компилятор установлен.

2.5.3 Использование устройства

Подготовка устройства к работе, подключение

1. Достаньте из коробки плату, USB кабель и CD-R диск с компилятором и драйверами.

2. Установите драйвера для JTAG-контроллера (PicoTAP) и интерфейсной микросхемы USB-UART (FT2232) (п. 2.5.2).

3. Проверьте положение перемычек (п. 2.5.1, рис. 4). Если у вас нет блока питания на 9-12 В, то плату при тестировании можно запитать от любого USB порта XS3 или XS6. Чтобы включить эту функцию, необходимо поставить джампер JMP5 из положения 1-2 в положение 2-3.

4. Подключаем интерфейсный провод USB к разъему XS3. На плате активируется светодиод наличия питания VD8, и ОС ПК найдет устройство PicoTAP.

Теперь плата готова к работе.

Проверка работоспособности платы

После того как устройство подготовлено к работе, проконтролируйте следующее:

- Светодиоды VD8 и VD10 должны гореть, при этом VD10 загорается позже, чем VD8. На LED индикаторе VD11 высветится число 0 и через ~1 с оно будет меняться на +1 (работает предзагруженное ПО в процессоре);

- С помощью вольтметра на контрольных точках измерьте напряжение: TP5 - +5 В $\pm 5\%$ (или 9-12 В), TP1 - +3.3 В $\pm 5\%$, TP2 - +1.8 В $\pm 5\%$;

- Нажмите и отпустите кнопку «reset», при нажатой кнопке VD10 и VD11 должны погаснуть. После того как кнопку отпустили, через некоторое время VD10 должен загореться (не более 1 с), а на VD11 снова начнется индикация. Если при проверке выявились какие-либо несоответствия, обратитесь к производителю платы.

Создание тестового примера программы

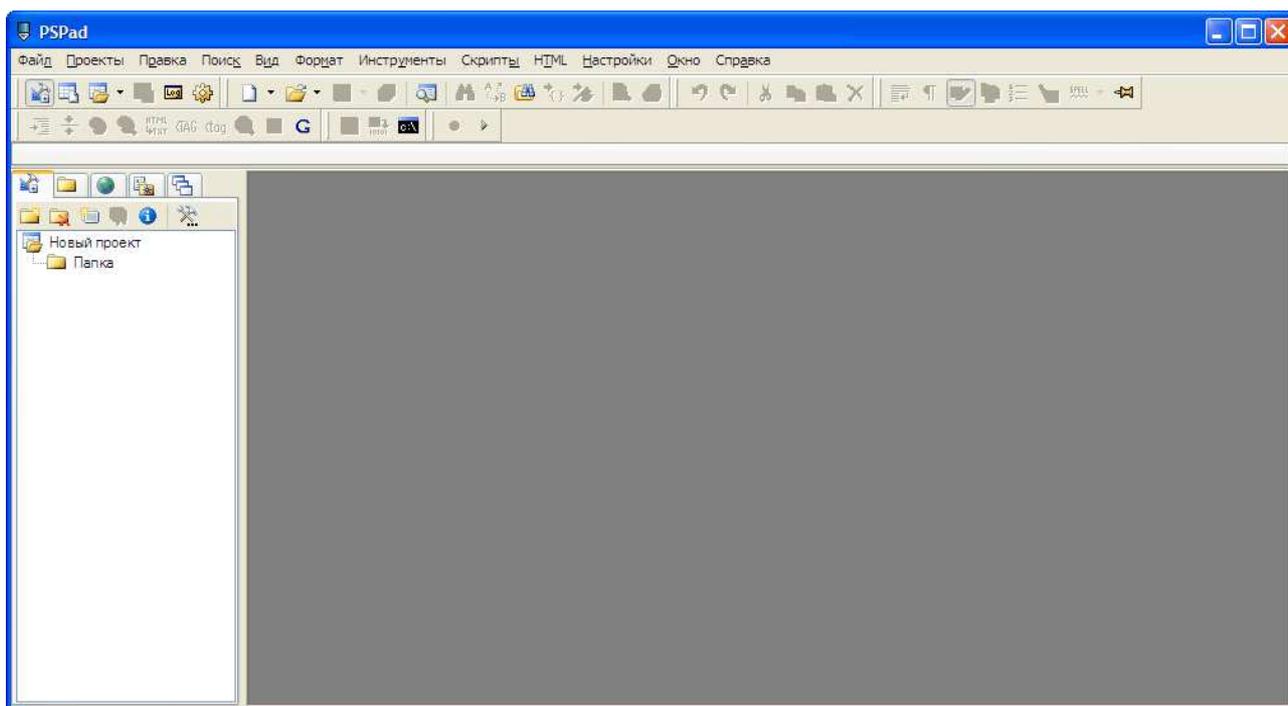
Чтобы быстро освоить азы работы с программно-аппаратным комплексом LDM-МСр0411100101-Q208 Evolution, приведем пошаговый пример по созданию простого программного кода.

ШАГ 1: Описание алгоритма работы ПО

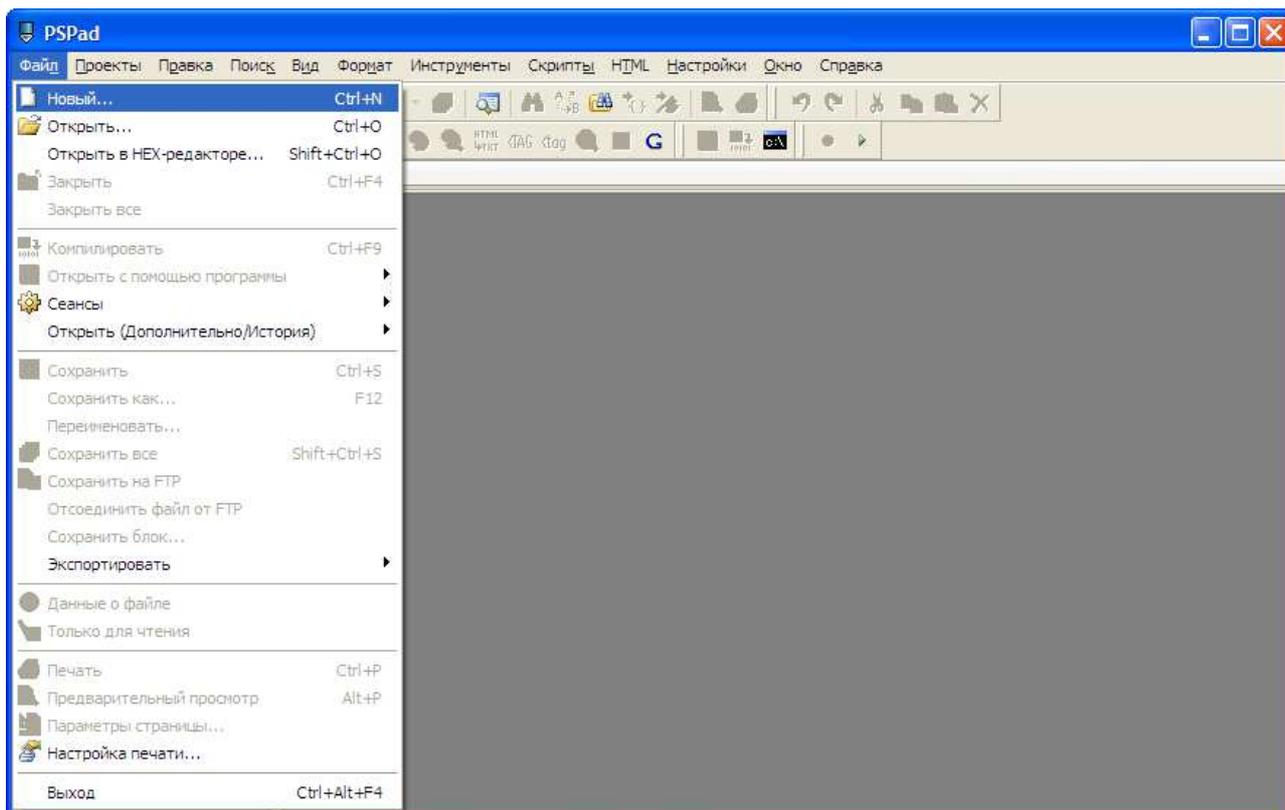
Индикатор VD11 отображает числа от 0 до 9. Увеличение значения числа на индикаторе производится при нажатии на кнопку SW2. Нажатие на кнопку SW1 приводит к уменьшению числа на индикаторе.

ШАГ 2: Создаем новый проект в системе MultiCletSDK

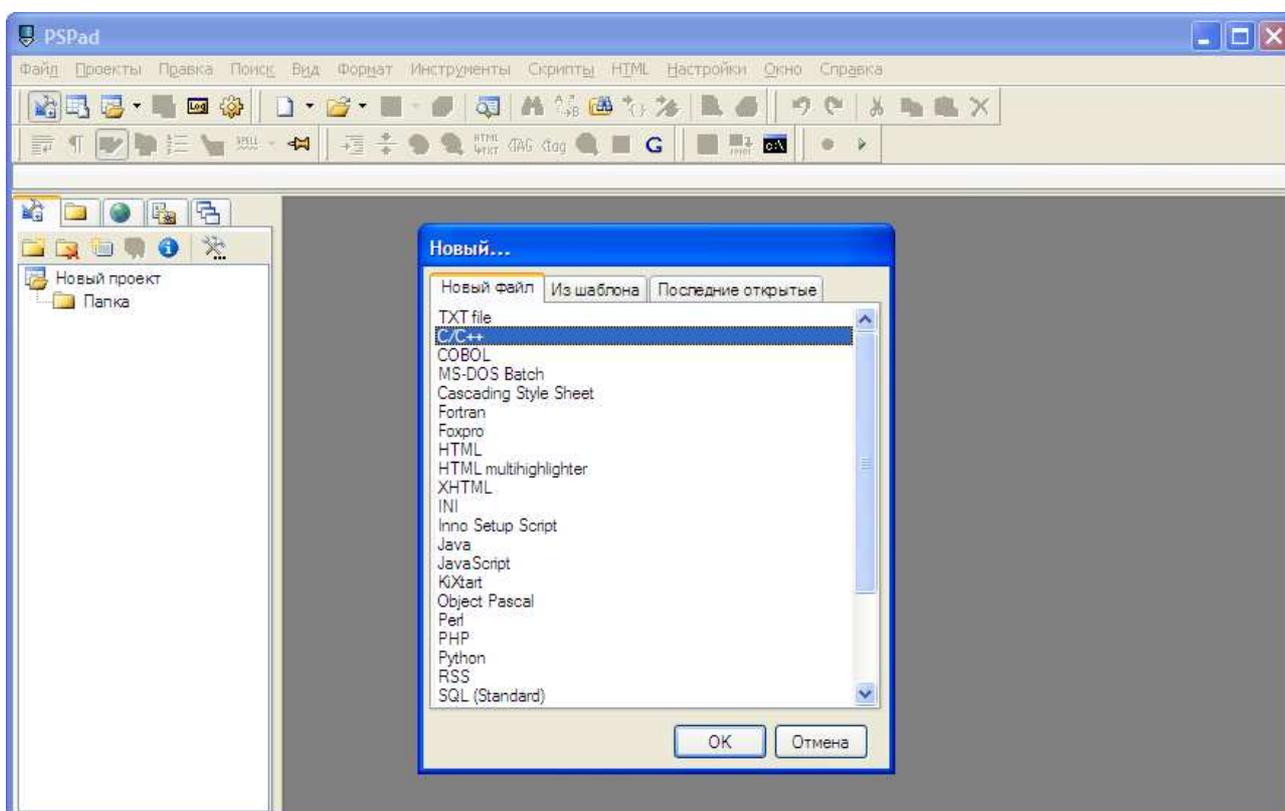
1) Заходим в меню Пуск/Все программы/MultiCletSDK/Редактор PSPad. Откроется окно PSPad:



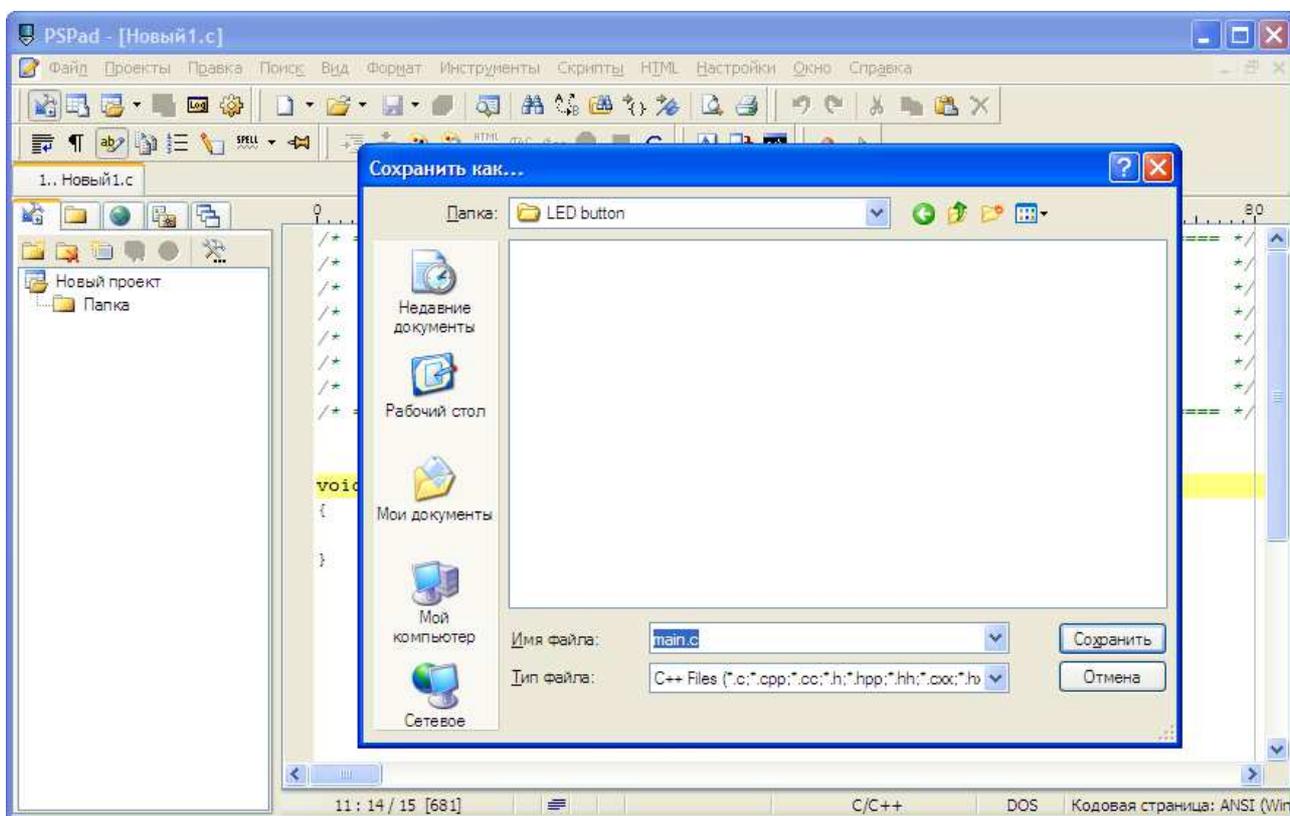
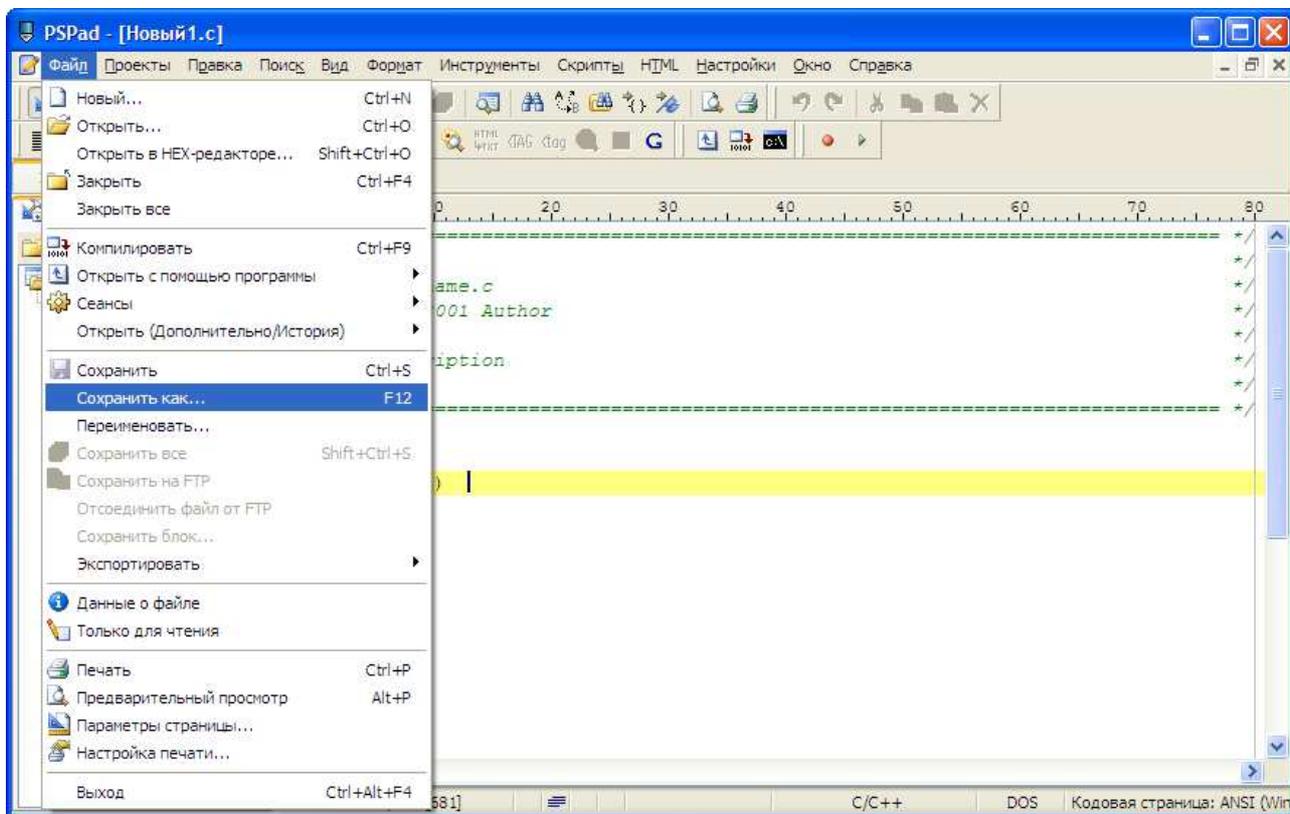
2) Заходим в раздел меню «Файл» и выбираем пункт «Новый».



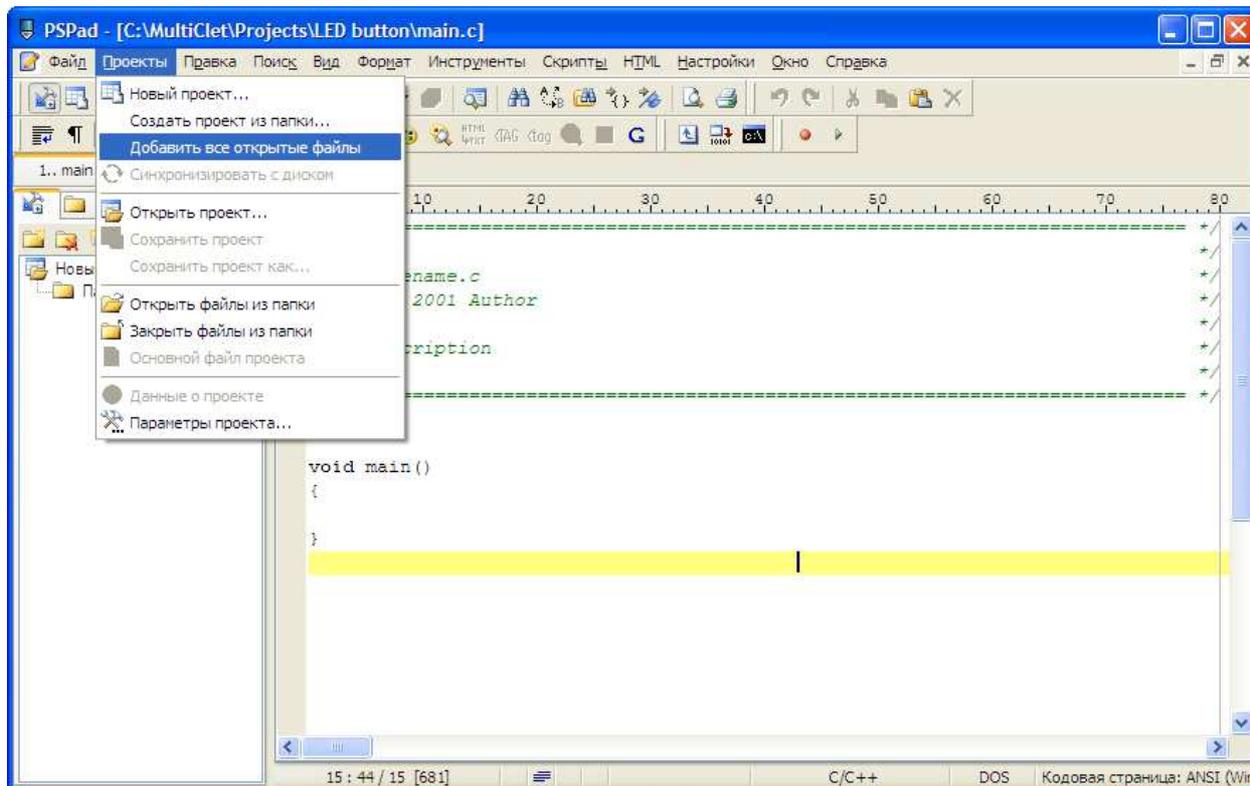
3) В открывшемся окне выбрать тип файла C/C++ и нажать ОК.



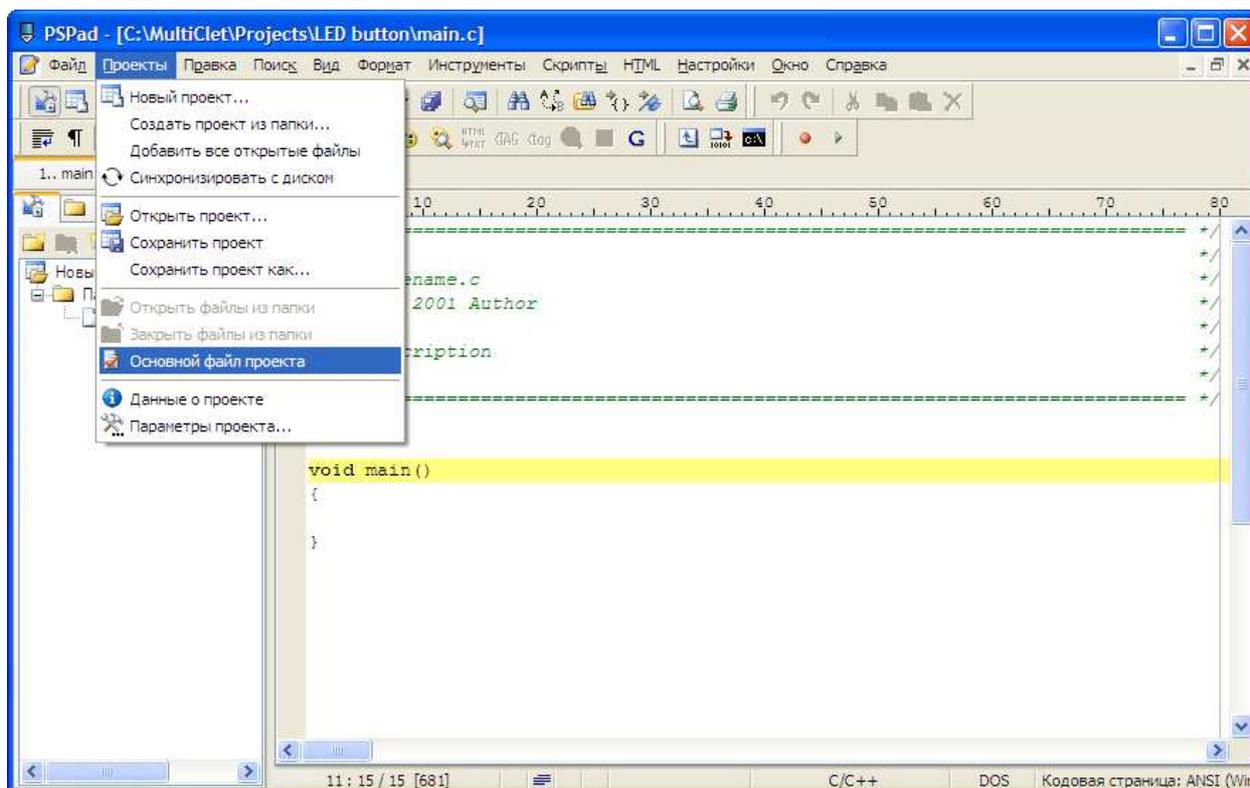
4) Произведем сохранение созданного файла. В разделе «Файл» выберем пункт «Сохранить как...». Создадим папку «LED button» в разделе «C:\MultiClet\Projects». Укажем имя файла «main.c» и нажмем кнопку «Сохранить».



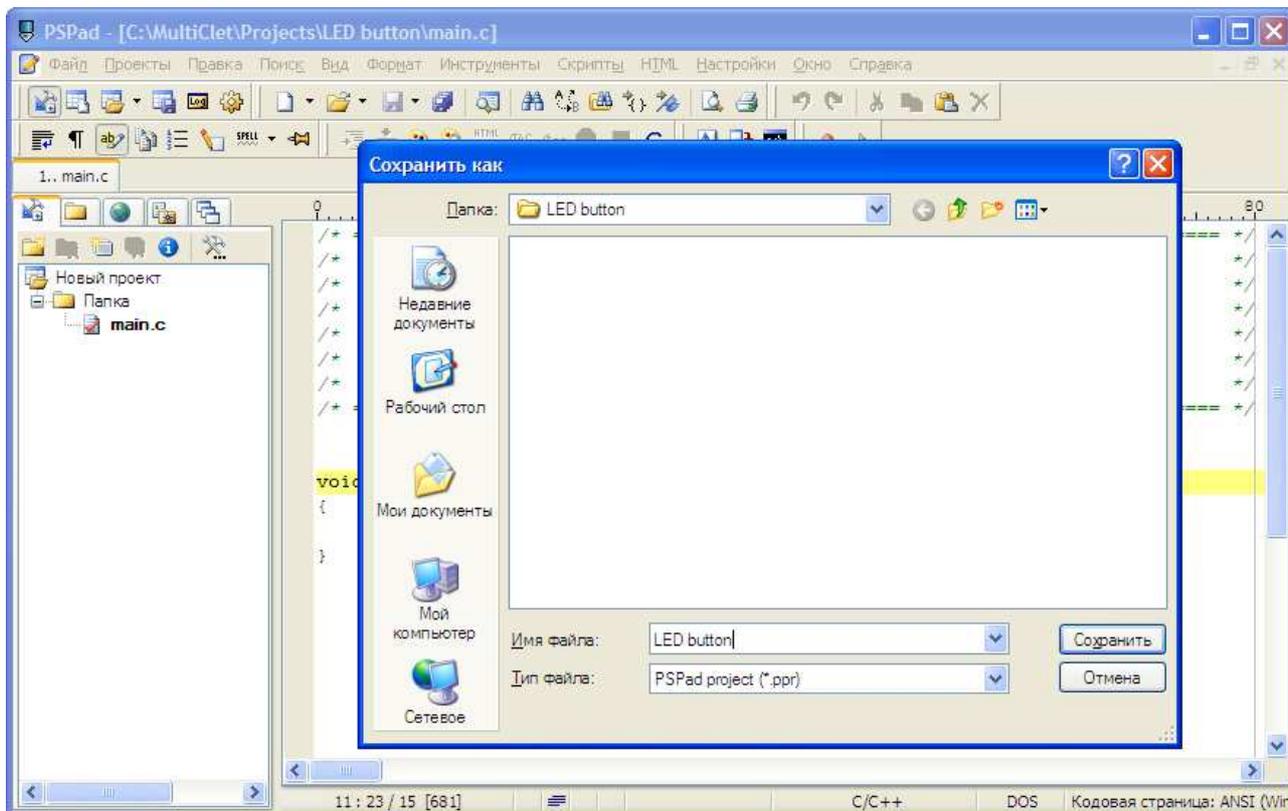
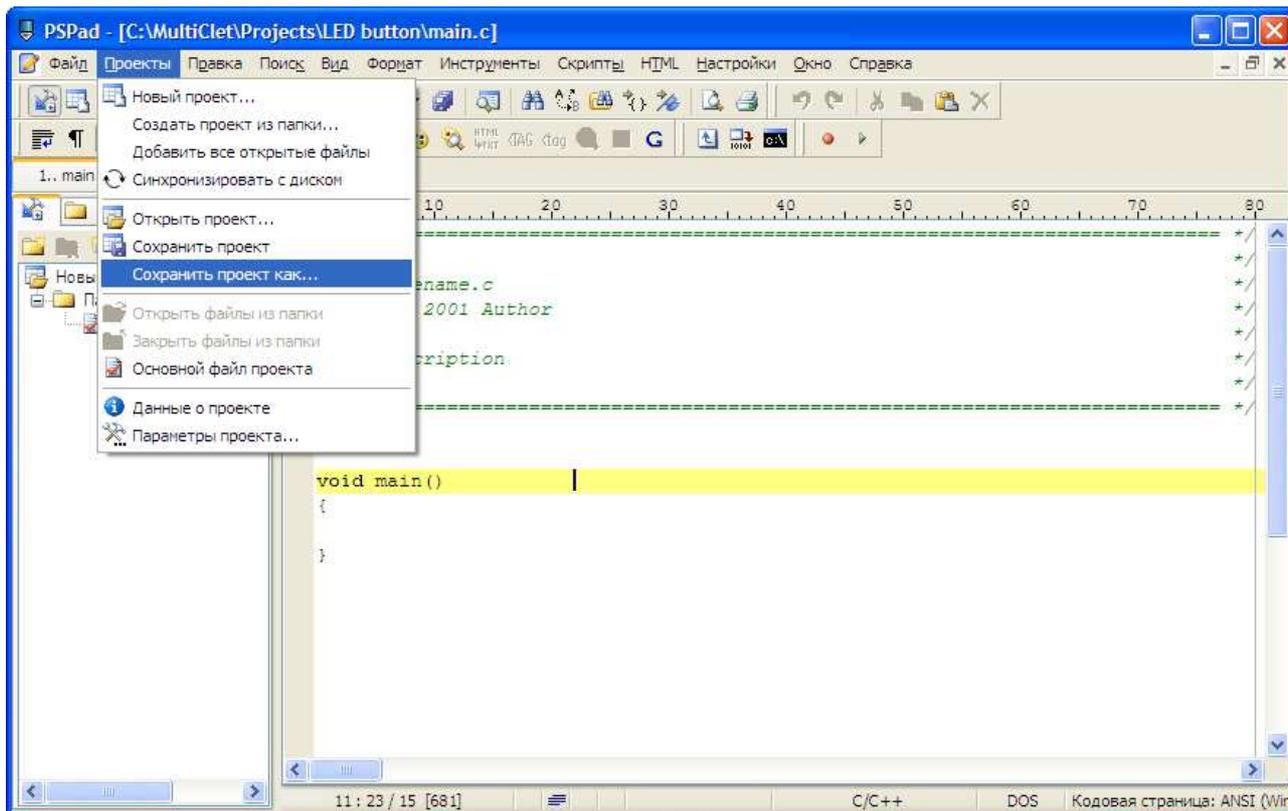
5) Добавим созданный файл к проекту. Выберем пункт меню «Проекты/Добавить все открытые файлы». После добавления файла в дереве проекта появится пункт «main.c».



6) Объявим файл main.c основным файлом проекта. Выделим файл main.c в дереве проекта и выполним действие «Проекты/Основной файл проекта».



7) Сохраним проект. Выберем «Проекты/Сохранить проект как...». Укажем имя проекта «LED button» и нажмем «Сохранить».



Проект подготовлен. В соответствующей папке проекта «LED button» созданы два файла: main.c, LED button.ppr.

ШАГ 3: Создаем директивы и структуры тестовой программы

1) Директивы типа операций

```
#define __I volatile const /* только чтение */
#define __O volatile /* только запись */
#define __IO volatile /* чтение/запись */
```

2) Тип переменной uint32_t

```
typedef unsigned int uint32_t;
```

3) Структура и директивы сторожевого таймера WDT

```
typedef struct
{
    __IO uint32_t CNT;
    __IO uint32_t KEY;
    __IO uint32_t RESERVED0;
    __IO uint32_t ST;
} WDT_TypeDef;

#define APB0PERIPH_BASE (0xC0000000)
#define WDT_BASE (APB0PERIPH_BASE + 0x000E0000)
#define WDT ((WDT_TypeDef *) WDT_BASE)
```

4) Структура и директивы портов В, С и D

```
typedef struct
{
    __IO uint32_t IN;
    __IO uint32_t OUT;
    __IO uint32_t DIR;
    __IO uint32_t MSK;
    __IO uint32_t POL;
    __IO uint32_t EDG;
    __IO uint32_t BPS;
} GPIO_TypeDef;

#define APB1PERIPH_BASE (0xC0100000)
#define GPIOB_BASE (APB1PERIPH_BASE + 0x000F0100)
#define GPIOB ((GPIO_TypeDef *) GPIOB_BASE)

#define GPIOC_BASE (APB1PERIPH_BASE + 0x000F0200)
#define GPIOC ((GPIO_TypeDef *) GPIOC_BASE)

#define GPIOD_BASE (APB1PERIPH_BASE + 0x000F0300)
#define GPIOD ((GPIO_TypeDef *) GPIOD_BASE)
```

5) Создаем директивы для удобной работы с портами вывода В и D

```
#define GPIOB_s(poz,val) if(val==1) GPIOB->OUT|=(1<<poz);else GPIOB->OUT&=~(1<<poz);
#define GPIOD_s(poz,val) if(val==1) GPIOD->OUT|=(1<<poz);else GPIOD->OUT&=~(1<<poz);
```

6) Создаем директивы линий LED дисплея

```
#define DA(x) GPIOB_s(25,x)
#define DB(x) GPIOB_s(26,x)
#define DC(x) GPIOB_s(27,x)
#define DD(x) GPIOB_s(28,x)
#define DE(x) GPIOB_s(29,x)
#define DF(x) GPIOB_s(30,x)
#define DG(x) GPIOB_s(31,x)

#define S1(x) GPIOB_s(20,x)
#define S2(x) GPIOB_s(21,x)
#define S3(x) GPIOB_s(23,x)
#define S4(x) GPIOB_s(24,x)
```

7) Создаем функцию временной задержки

```
void Delay(int data)
{
    int j,k;
    for(j=0; j<data; j++)
        for(k=0; k<1000; k++);
}
```

8) Создаем функцию формирования чисел на LED дисплее

```
void DataL(unsigned char data)
{
    switch(data)
    {
        case 0:
            DA(0); DB(0); DC(0); DD(0); DE(0); DF(0); DG(1);
            break;

        case 1:
            DA(1); DB(0); DC(0); DD(1); DE(1); DF(1); DG(1);
            break;

        case 2:
            DA(0); DB(0); DC(1); DD(0); DE(0); DF(1); DG(0);
            break;

        case 3:
            DA(0); DB(0); DC(0); DD(0); DE(1); DF(1); DG(0);
            break;
```

```

case 4:
    DA(1); DB(0); DC(0); DD(1); DE(1); DF(0); DG(0);
    break;

case 5:
    DA(0); DB(1); DC(0); DD(0); DE(1); DF(0); DG(0);
    break;

case 6:
    DA(0); DB(1); DC(0); DD(0); DE(0); DF(0); DG(0);
    break;

case 7:
    DA(0); DB(0); DC(0); DD(1); DE(1); DF(1); DG(1);
    break;

case 8:
    DA(0); DB(0); DC(0); DD(0); DE(0); DF(0); DG(0);
    break;

case 9:
    DA(0); DB(0); DC(0); DD(0); DE(1); DF(0); DG(0);
    break;
}
}

```

ШАГ 4: Корректируем внутреннюю структуру функции main() {}

1) Добавляем переменные

```

int status_in; // Переменная состояния линий In порта C
unsigned char Led_dat = 0; // Номер символа LED дисплея

```

2) Выключаем WDT

```

WDT->KEY = ((uint32_t)0x00003333); // Выключаем WDT

```

3) Настраиваем выходы портов на прием и передачу

```

GPIOC->DIR = ((uint32_t)(0<<23)|(0<<22)|(0<<21)|(0<<20));
GPIOD->DIR = ((uint32_t)(1<<12)|(1<<13)|(1<<14)|(1<<15));
GPIOB->DIR = ((uint32_t)0xFFB00000);

```

4) Включаем первый разряд LED дисплея

```

S1(0); S2(0); S3(0); S4(1);

```

5) Устанавливаем начальное значение символа

```

DataL(Led_dat);

```

6) Добавляем основной блок программы

```
while(1)
{
    status_in = GPIOC->IN; // Запоминаем значения регистра IN порта C

    if(!(status_in & (1<<20))) // Если нажата кнопка SW1, то уменьшаем значение LED
    {
        if(Led_dat > 0)
            Led_dat--;
        else
            Led_dat = 9;

        DataL(Led_dat);

        GPIOD_s(12,0); // Зажигаем светодиод VD3 рядом с SW1
        Delay(1000);
    }
    else if(!(status_in & (1<<21))) // Если нажата кнопка SW1, то уменьшаем значение LED
    {
        if(Led_dat < 9)
            Led_dat++;
        else
            Led_dat = 0;

        DataL(Led_dat);

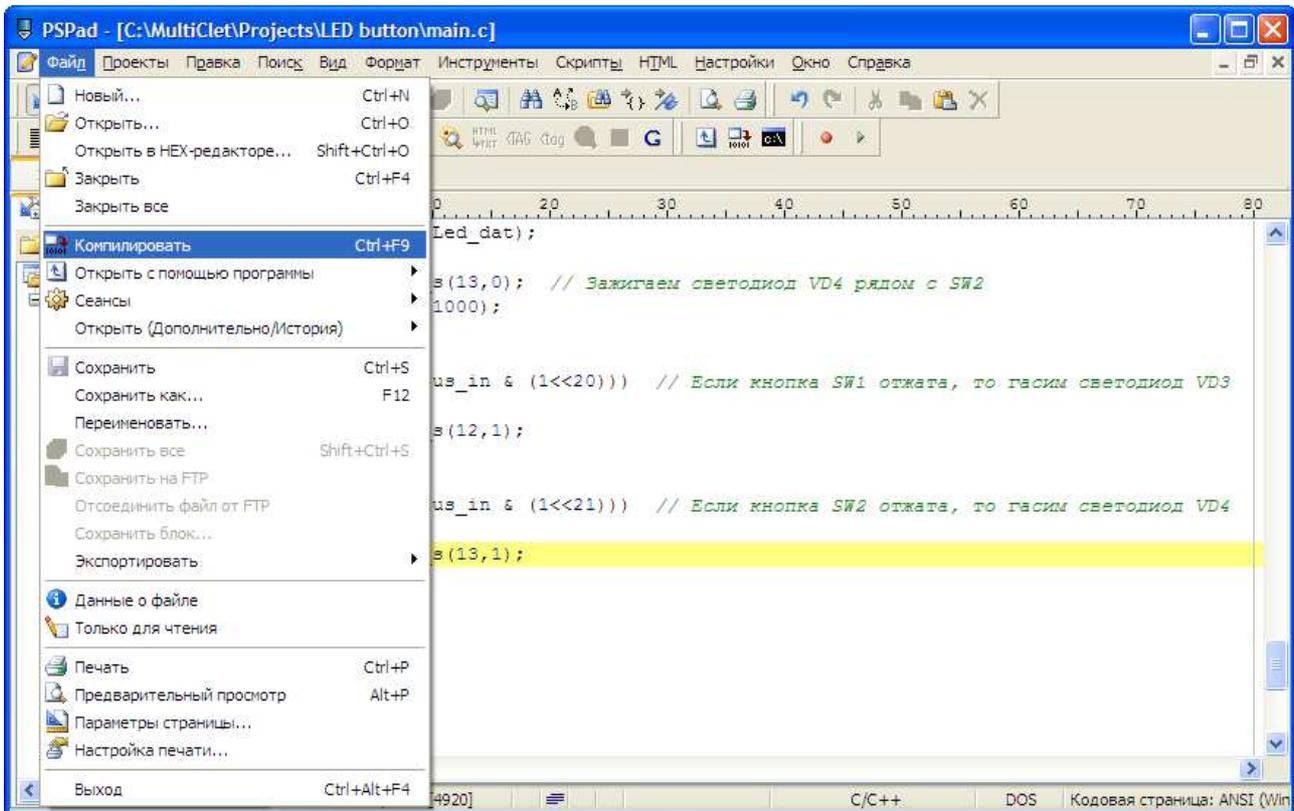
        GPIOD_s(13,0); // Зажигаем светодиод VD4 рядом с SW2
        Delay(1000);
    }

    if((status_in & (1<<20))) // Если кнопка SW1 отжата, то гасим светодиод VD3
    {
        GPIOD_s(12,1);
    }

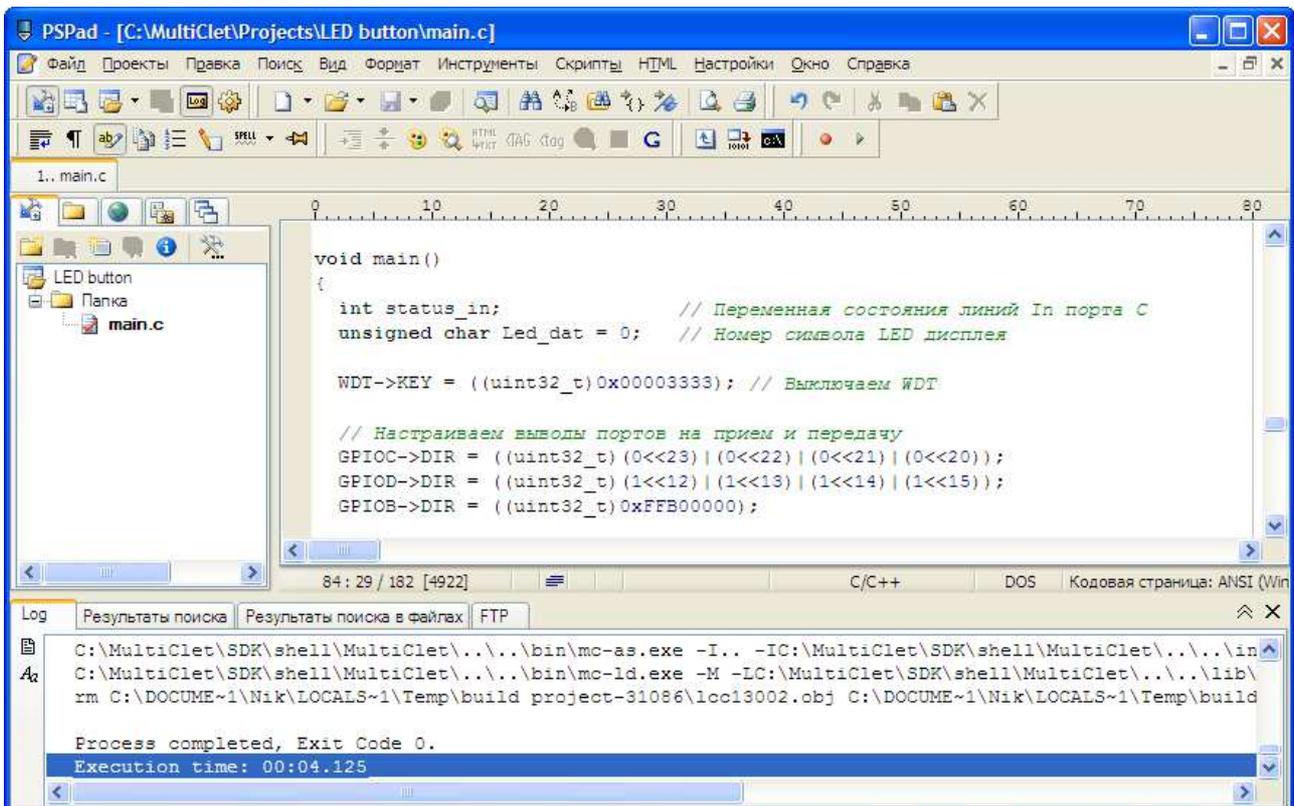
    if((status_in & (1<<21))) // Если кнопка SW2 отжата, то гасим светодиод VD4
    {
        GPIOD_s(13,1);
    }
}
```

ШАГ 5: Компиляция проекта

Выбираем пункт меню «Файл/Компилировать» или нажимаем Ctrl+F9.



События процесса компиляции отображаются в открывшемся поле внизу окна.



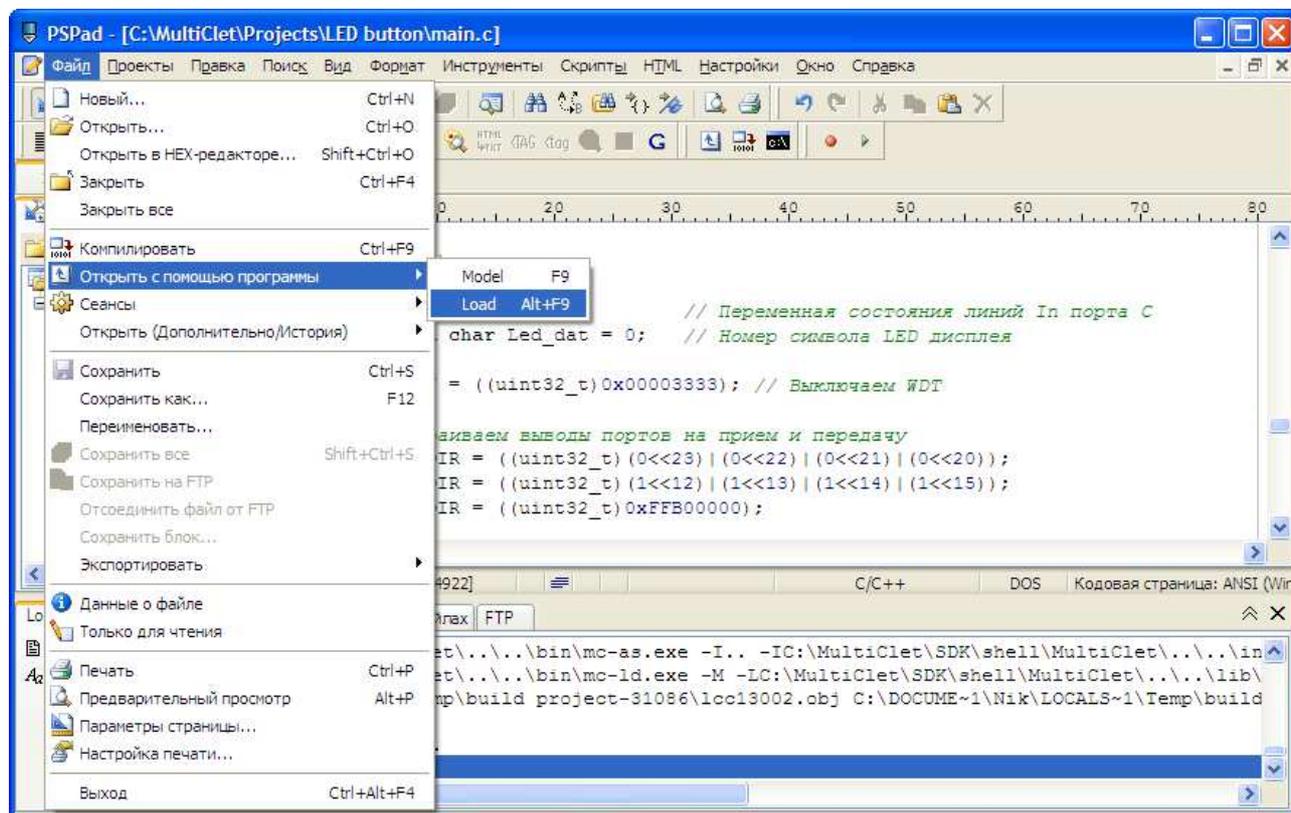
Ожидаем окончания процесса компиляции:

Process completed, Exit Code 0.
Execution time: 00:04.125

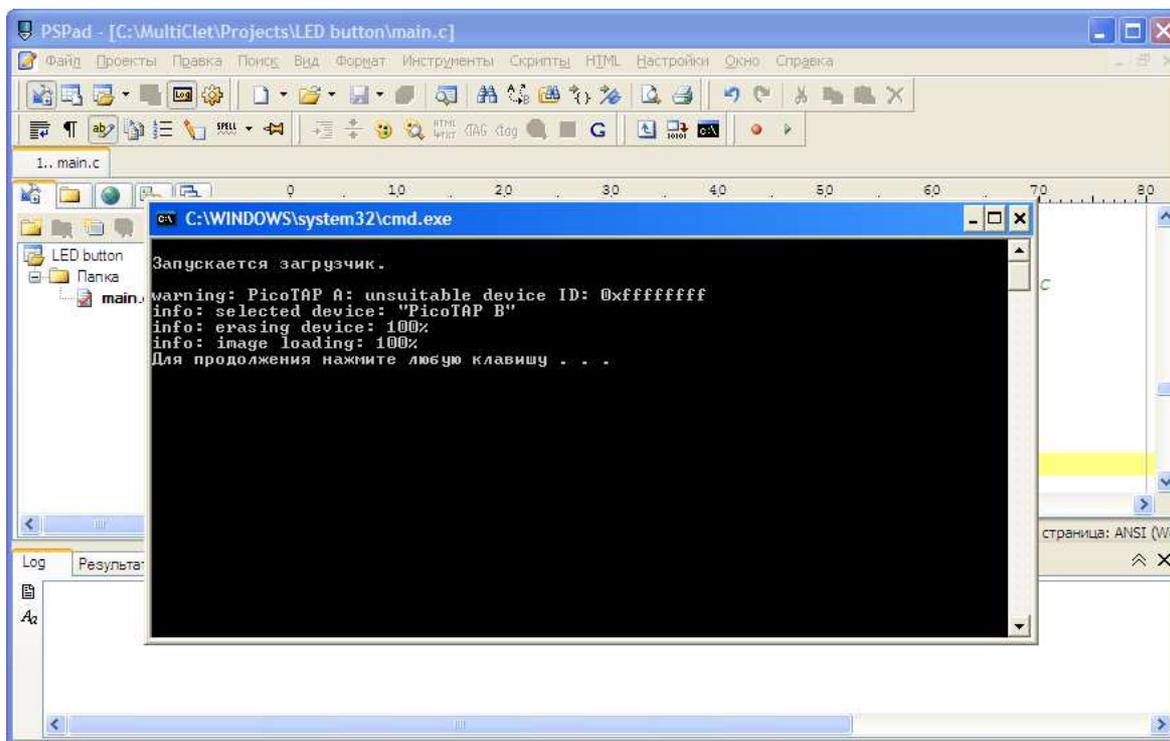
В каталоге проекта появится папка «out», в которой будет размещен бинарный файл «image.bin».

Загрузка проекта на плату

Проверьте подключение платы к USB порту ПК и наличие питания по светодиоду VD8. Запускаем процесс загрузки прошивки на плату. Выбираем пункт меню «Файл/Открыть с помощью программы/Load» или используем комбинацию клавиш Alt+F9.



Откроется дополнительное окно, в котором будет отображаться процесс стирания памяти и загрузки образа проекта:



После завершения загрузки образа необходимо нажать на кнопку сброса Reset SW5. О завершении загрузки прошивки в процессор оповестит включенный светодиод VD10. Теперь можно проверить работоспособность программы.

На LED дисплее высветится число «0».

Произведем однократное нажатие на кнопку SW2. Значение на индикаторе изменится на «1». При последующих нажатиях на SW2 будет производиться индикация увеличивающихся значений чисел до «9». При последующем нажатии на кнопку SW2 на индикаторе отобразится «0», и алгоритм будет работать в аналогичном цикле.

Если нажимать кнопку SW1, то числа на индикаторе будут уменьшаться. Будет формироваться переход значения индикатора в «9» при переходе через «0».

Если вы освоили данный раздел, то теперь вы можете создать свой алгоритм и опробовать различные интерфейсы: UART, USB, I2S, I2C, SPI и другие периферийные блоки процессора и самой платы: АЦП, ЦАП, EEPROM и др.

2.6 Принципиальные электрические схемы отладочной платы

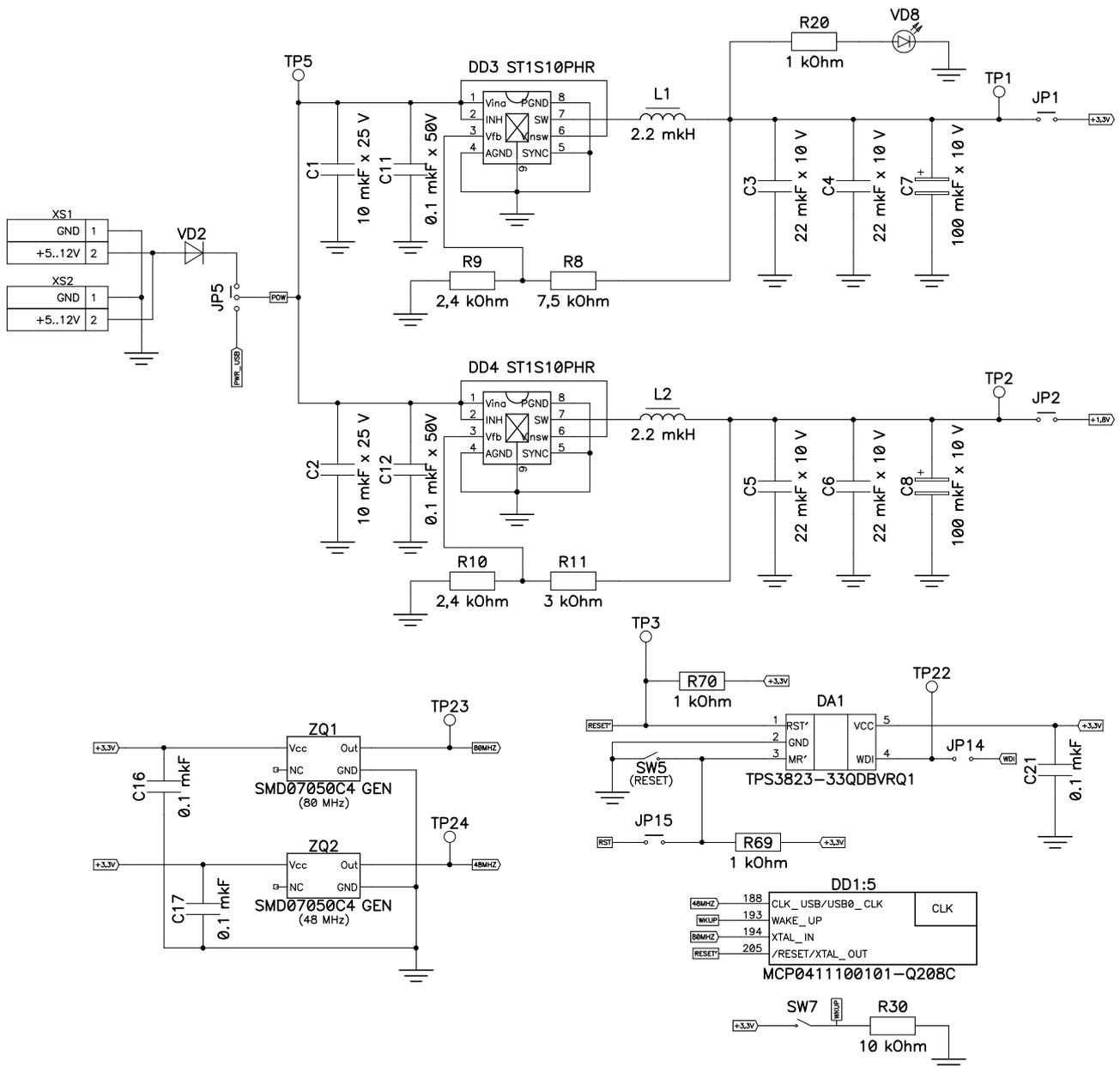


Рисунок 3. Принципиальная электрическая схема преобразователей напряжения 3,3 В и 1,8 В

Для удобства контроля электрических параметров и формы сигналов на плате размещены контрольные точки (TPx) для всех основных компонентов и линий питания платы.

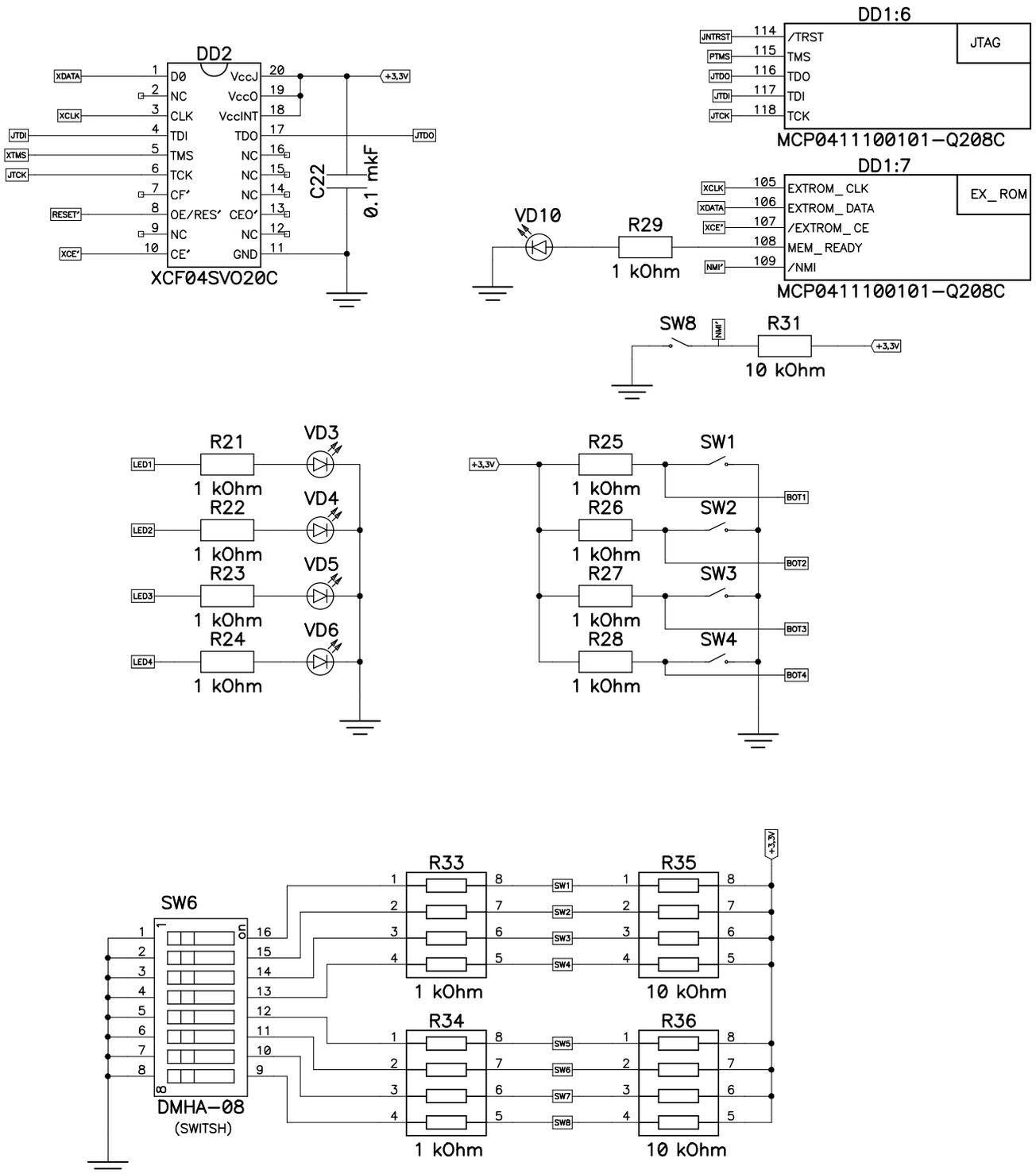


Рисунок 4. Принципиальная электрическая схема FLASH ПЗУ, загрузочный блок процессора, порты I/O

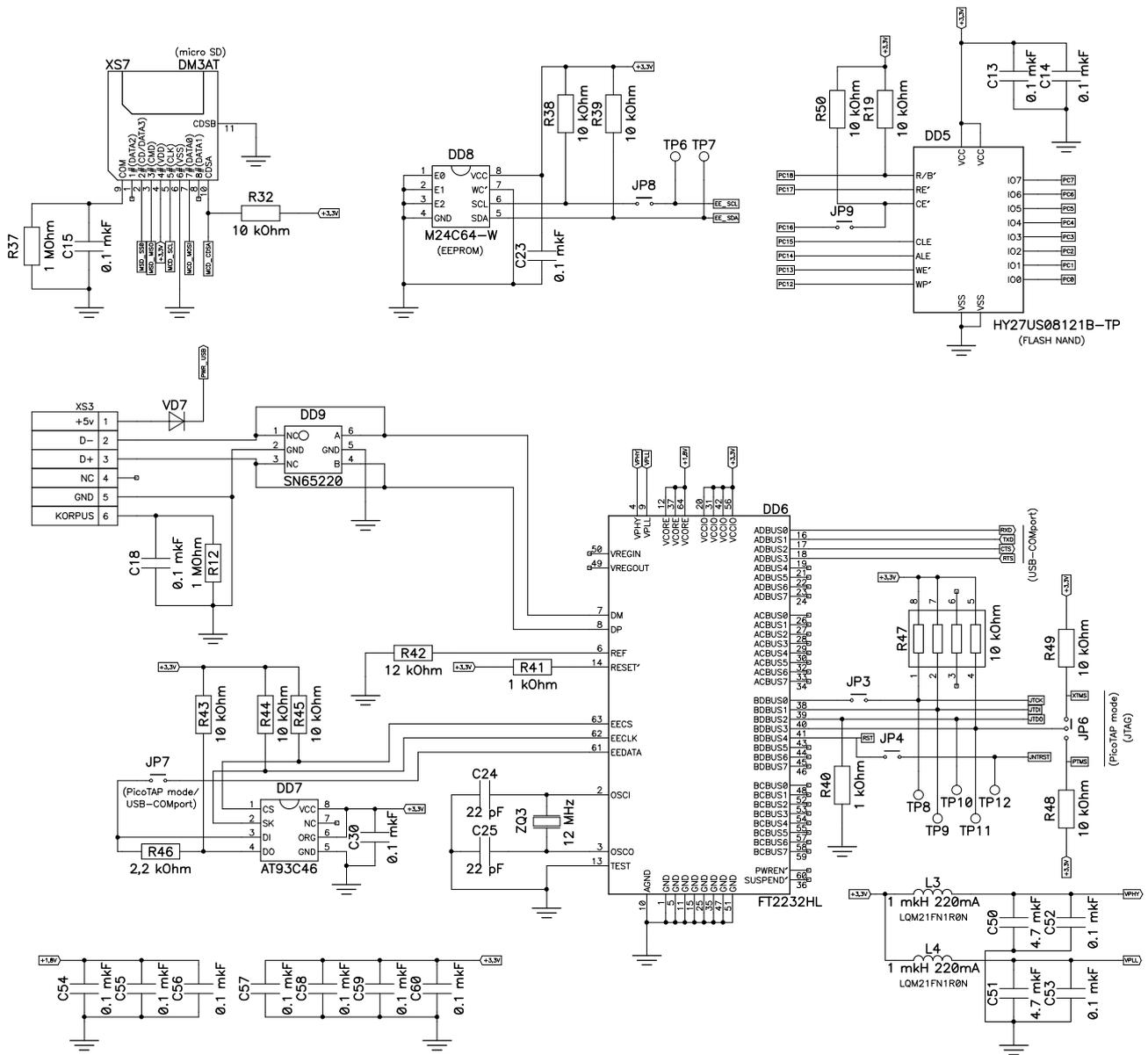


Рисунок 5. Принципиальная электрическая схема
micro SD карта, EEPROM, NAND FLASH, загрузчик PicoTAP + USB-COM порт

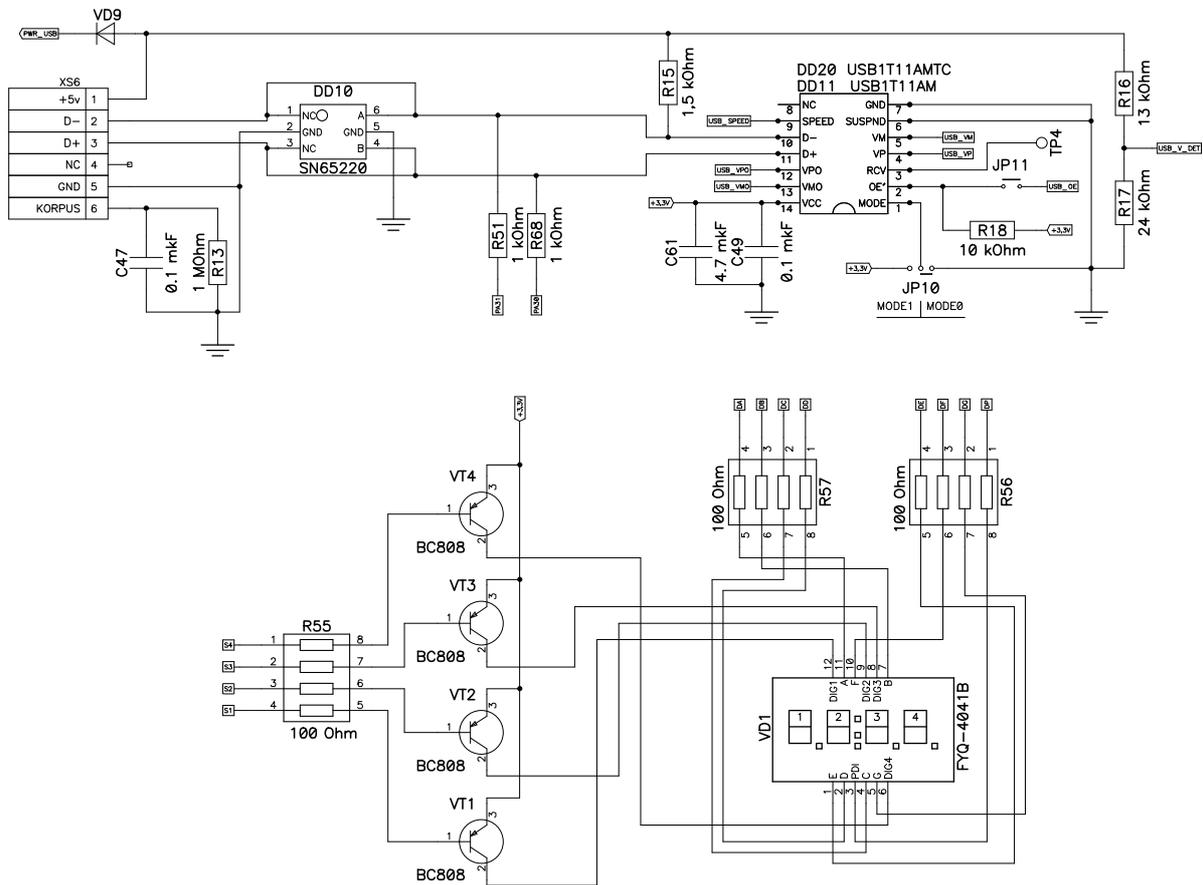


Рисунок 6. Принципиальная электрическая схема USB порт, LED дисплей 4 знака

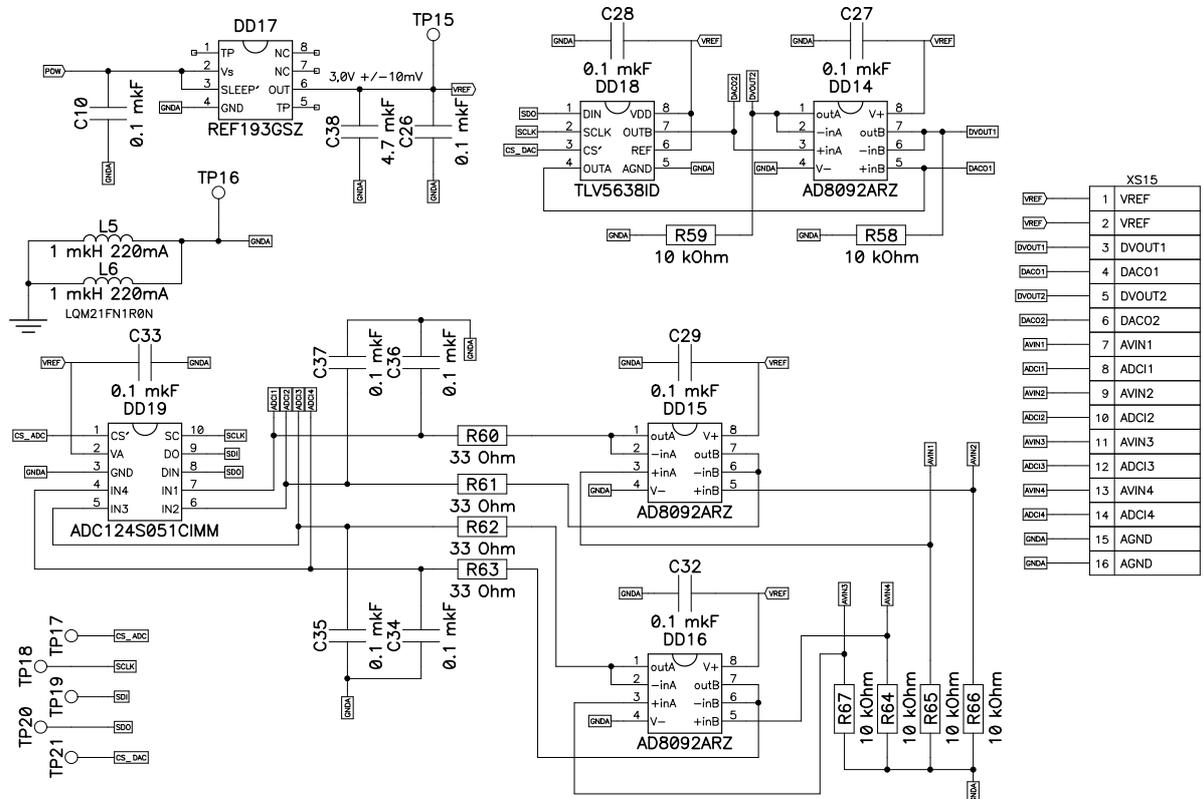


Рисунок 7. Принципиальная электрическая схема АЦП 4 канала 12 бит, ЦАП 2 канала 12 бит

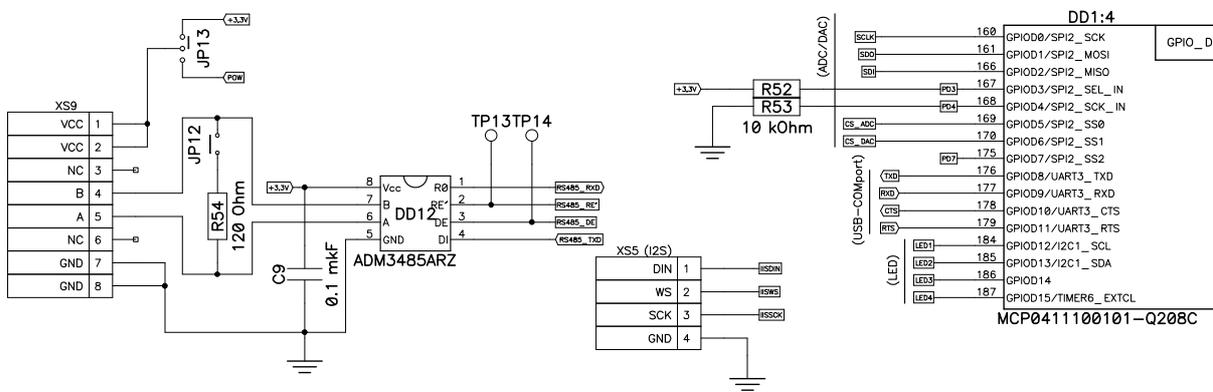
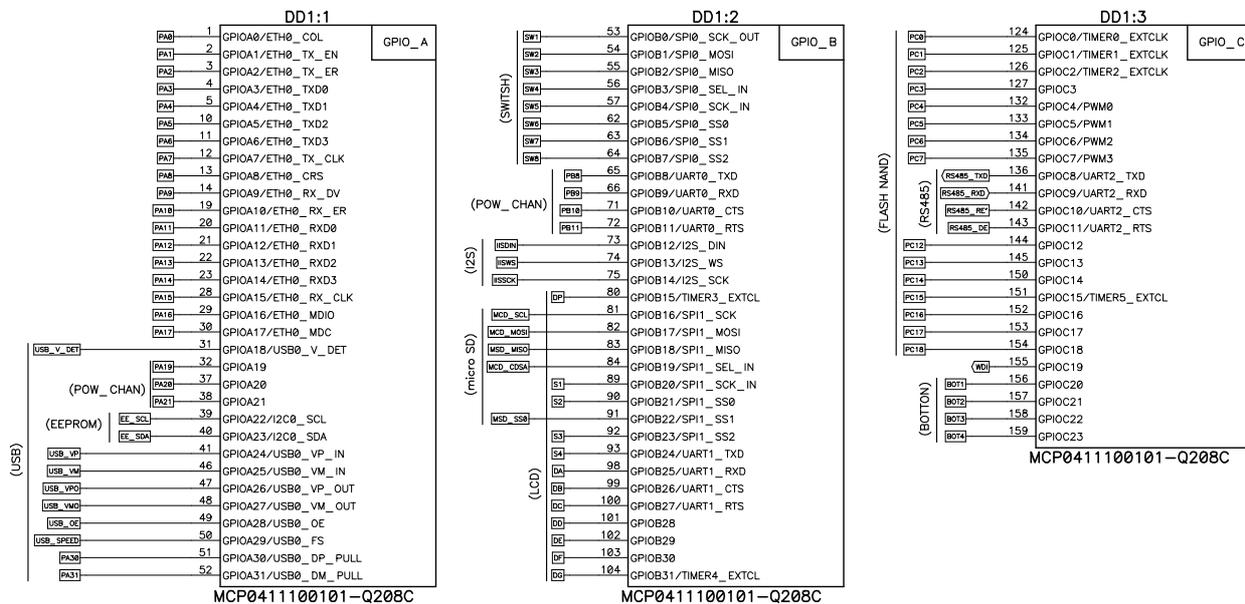


Рисунок 8. Принципиальная электрическая схема порты ввода/вывода процессора, интерфейсы RS485 и I2S

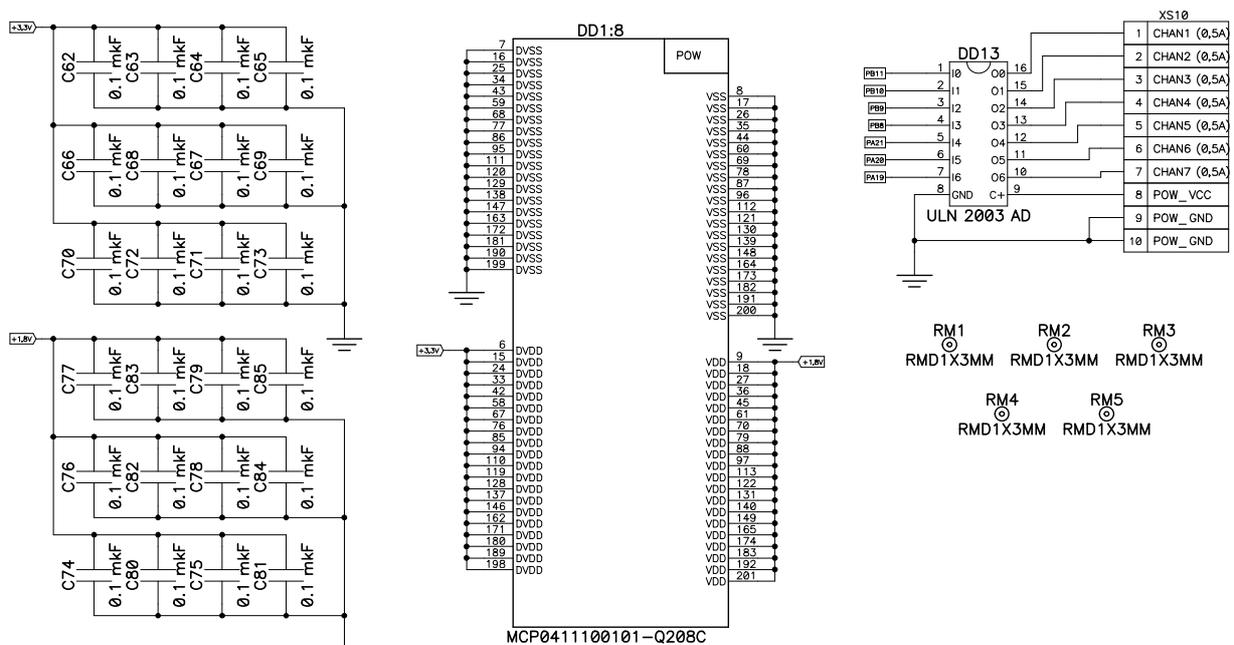


Рисунок 9. Принципиальная электрическая схема подключения напряжения к выводам процессора, силовых выходов (7 каналов 0,5А, 30В)

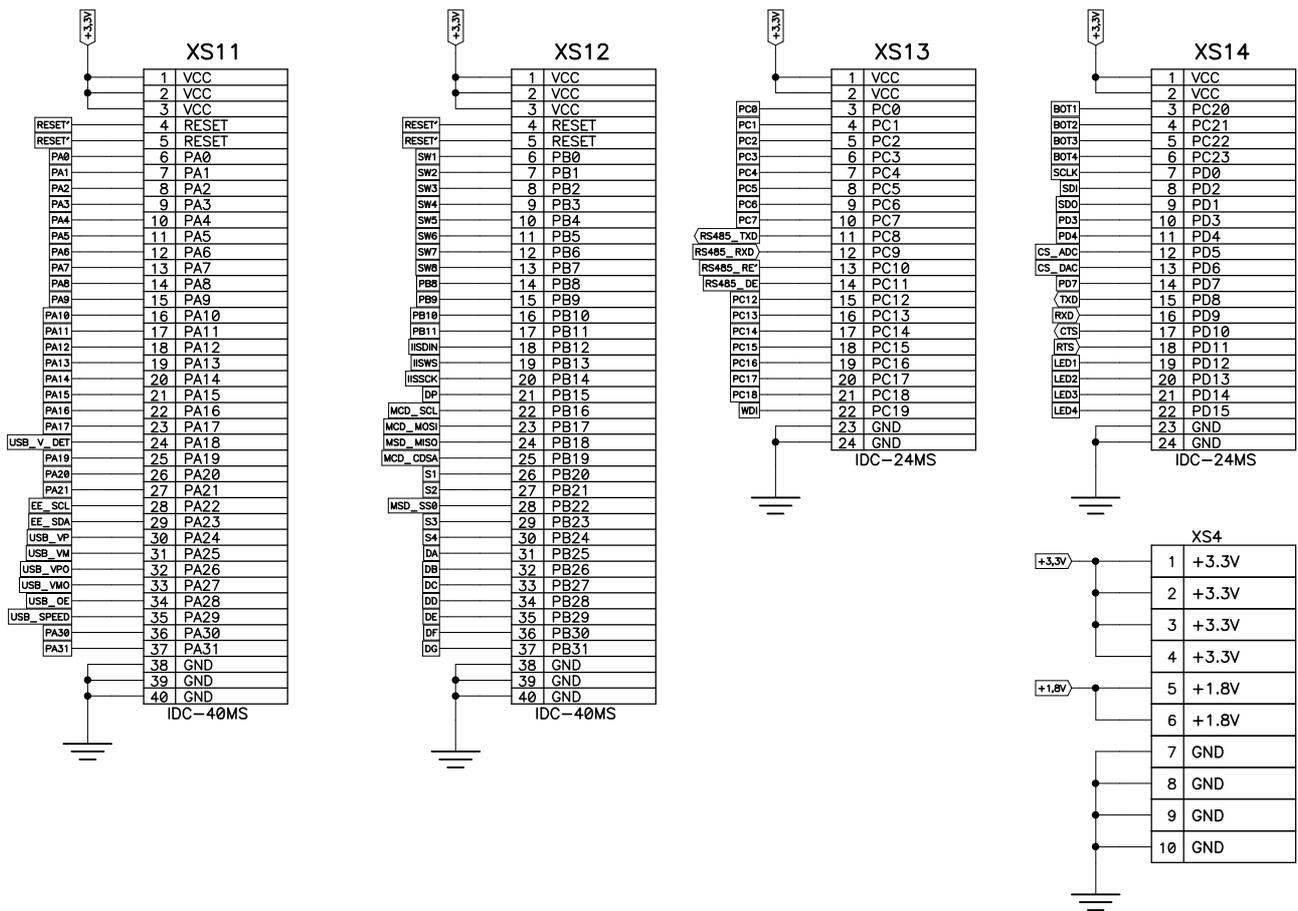
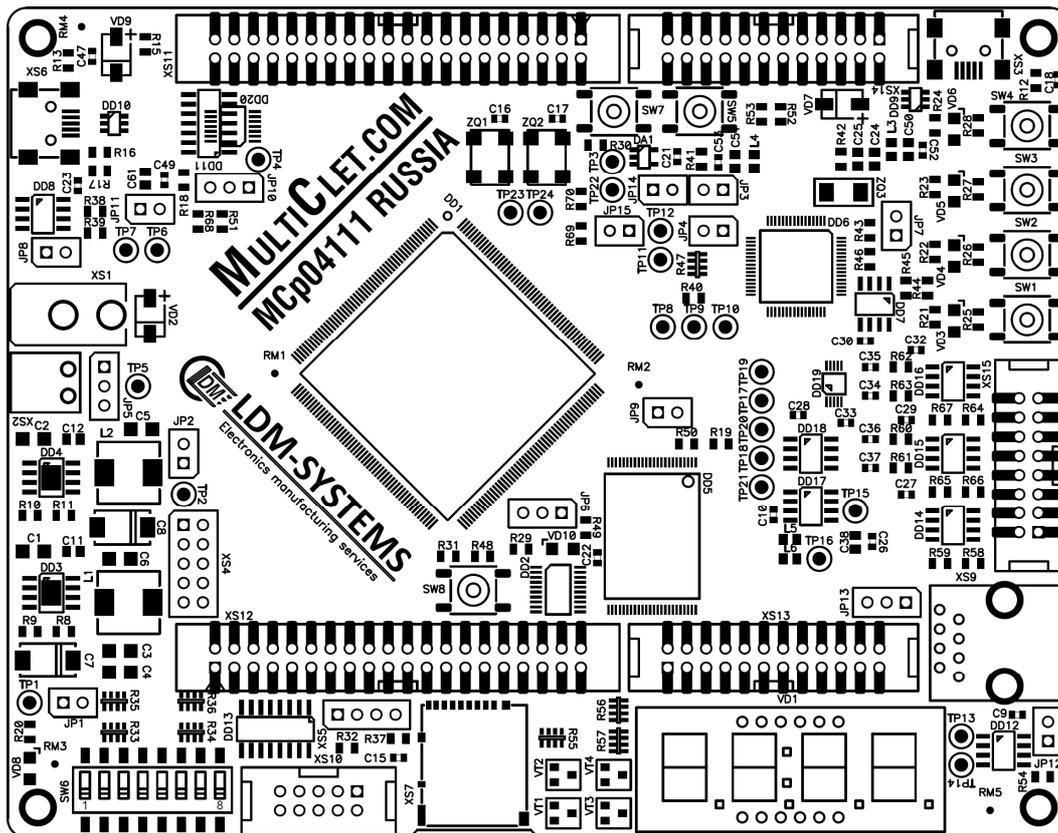


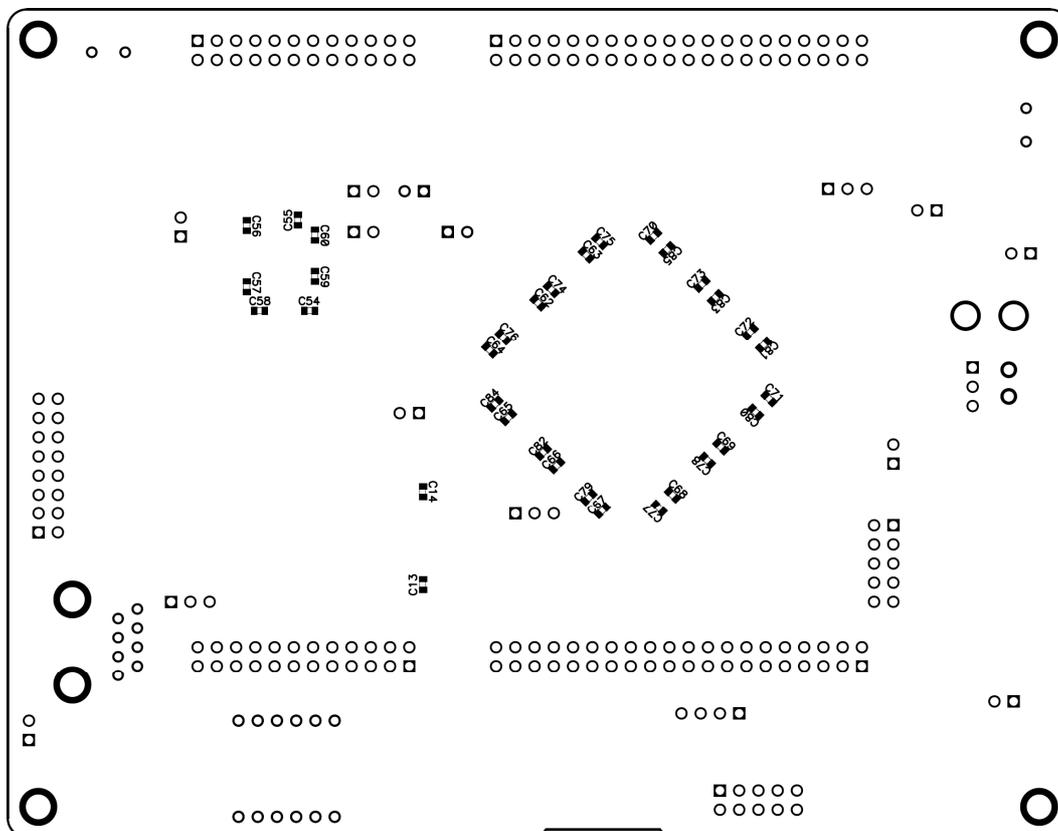
Рисунок 10. Принципиальная электрическая схема внешних разъемов

2.7 Монтажные чертежи

Слой TOP

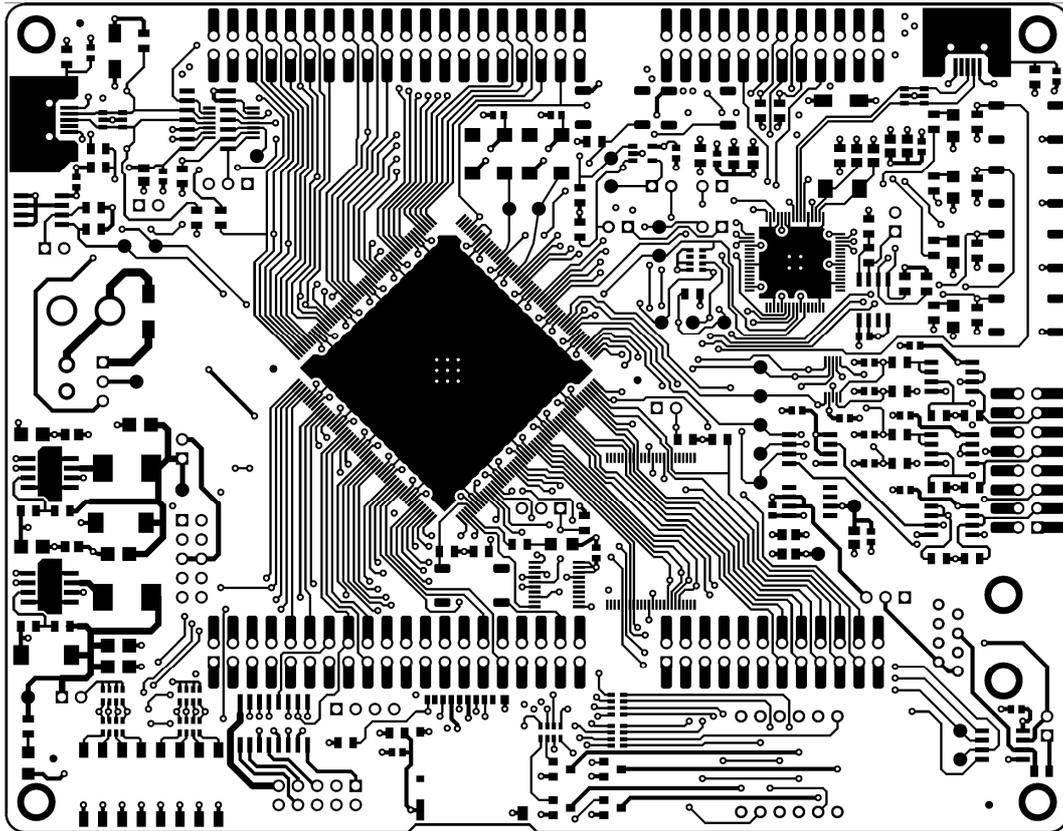


Слой BOTTOM

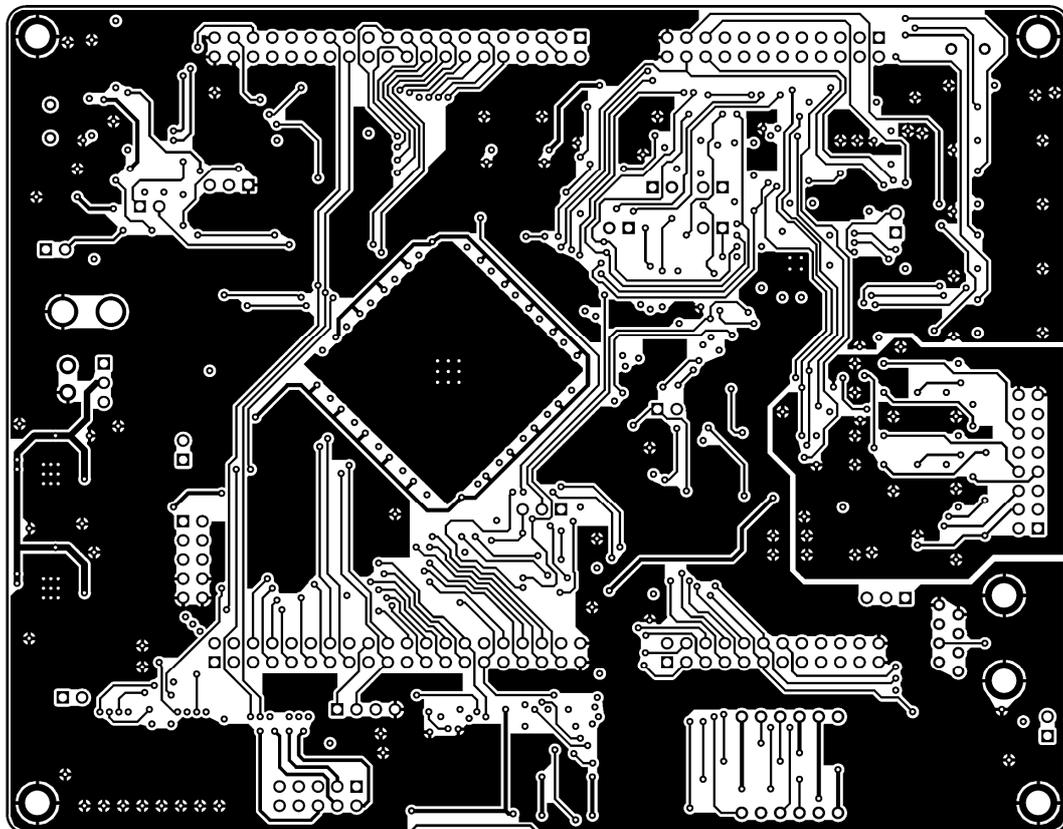


2.8 Трассировка по слоям

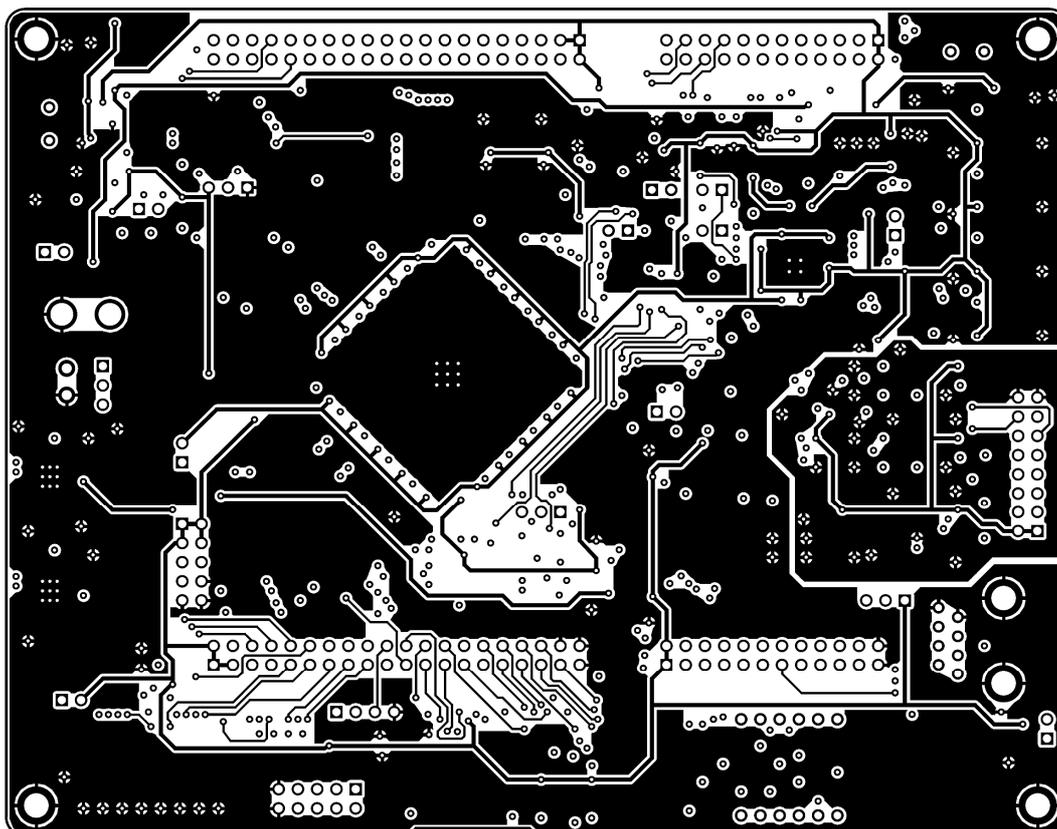
Слой TOP



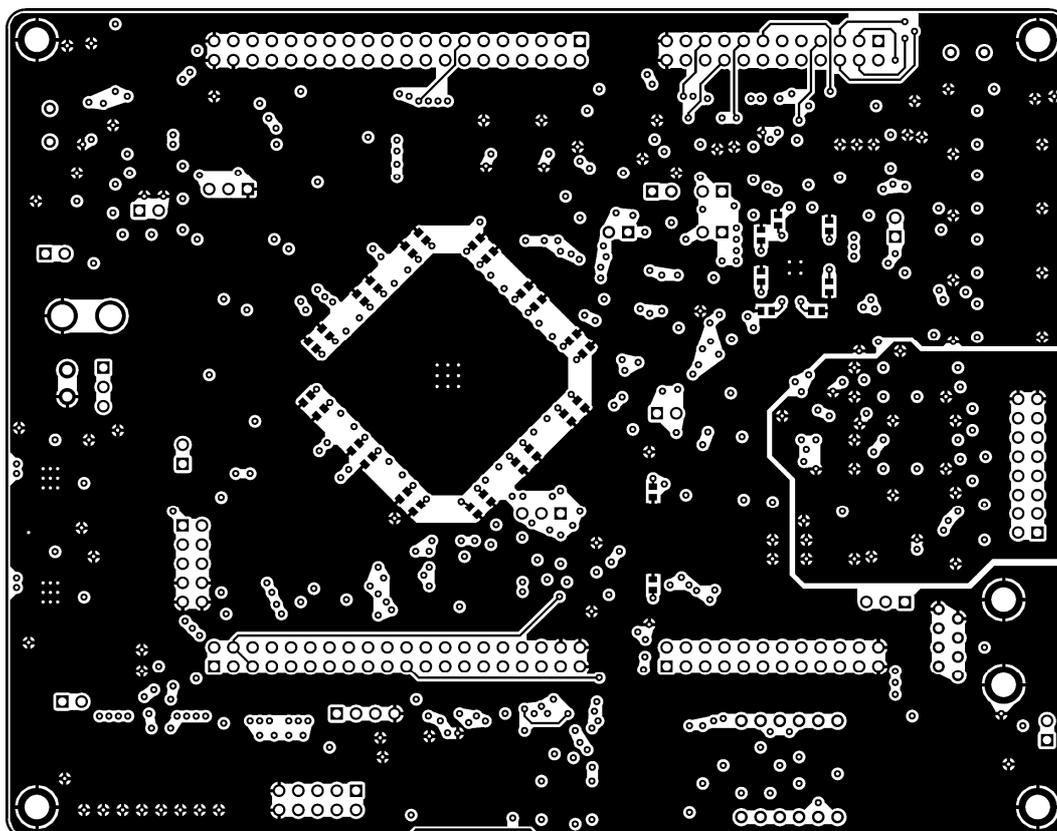
Слой LAYER1



Слой LAYER2



Слой BOTTOM



3 ЭКСПЛУАТАЦИЯ, ХРАНЕНИЕ И ТРАНСПОРТИРОВАНИЕ

Требования к условиям эксплуатации:

Изделие при испытаниях, перевозке, хранении и эксплуатации не наносит вреда окружающей среде и здоровью человека. Сохраняет свои параметры во всем диапазоне рабочих температур от 0°C до +70°C в закрытом помещении с относительной влажностью воздуха не более 80%, без конденсата, при изменении напряжения первичного источника электропитания в допустимых пределах. По электромагнитной совместимости изделие соответствует всем требованиям для аппаратуры данного класса.

Требования к условиям хранения:

Изделие должно храниться в складских помещениях, защищенных от воздействий атмосферных осадков, на стеллажах в упаковке производителя при отсутствии в воздухе паров кислот, щелочей и других веществ, вызывающих коррозию. Условия хранения изделия по ГОСТ 15150-69: температура воздуха от +5°C до +40°C, относительная влажность до 80% при температуре +25°C. Предельный срок хранения в указанных условиях - три года.

Требования к условиям транспортирования:

Транспортирование изделия разрешается в упаковке производителя всеми видами транспорта, за исключением негерметизированных отсеков самолета, без ограничения расстояния.

Транспортирование упакованных изделий может производиться в крытых вагонах и автомашинах, трюмах судов и герметичных кабинах самолетов при температуре воздуха от -20°C до +70°C. При любом способе транспортирования необходимо предусмотреть крепление ящика к кузову (платформе) транспортного средства с помощью крепежной арматуры.