LDM-SYSTEMS

E-mail: info@ldm-systems.com
URL: www.ldm-systems.com
tel.: 007 (495) 500-89-20
fax: 007 (495) 536-46-67

SERVICE MANUAL

Development Board
LDM-MCp0411100101-Q208 Evolution

Made in Russia

LDM-SYSTEMS
January 2013

Contents

INTRODUCTION

LDM-MCp0411100101-Q208 Evolution development kit is designed for high-performance, power efficient devices development based on multi-cell MCp0411100101-Q208I processor of a Russian company JSC “MultiClet”. Well-balanced board architecture allows to simplify the elaboration and implementation programming modules as long as minimize financial cost and schedule times. This development kit is designed to meet both beginners’ and professionals’ needs.

This manual describes basic characteristics and operation of LDM-MCp0411100101-Q208 Evolution (fig. 1); drivers and software setup is also presented. The manual contents in brief main features, architecture and periphery of MCp0411100101 microprocessor.

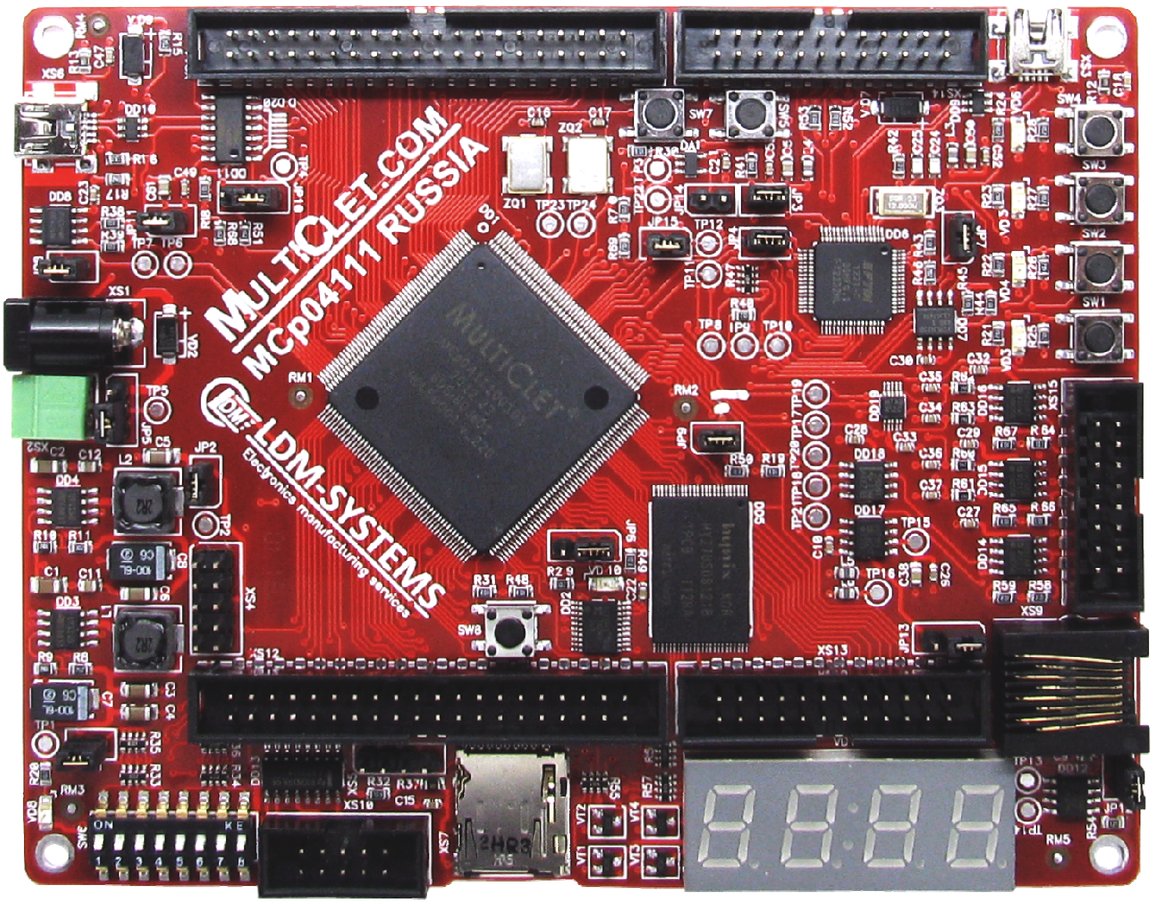


Figure 1. Main view of LDM-MCp0411100101-Q208 Evolution

1 ABBREVIATIONS

MP – Microprocessor;

ROM – Read-only memory;

PM – Program memory;

SW – Software;

DM – Data memory.

2 DESCRIPTION and OPERATION

2.1 Device application

LDM-MCp0411100101-Q208 Evolution Development kit is designed for development systems based on multi-cell MCp0411100101 processor made by Multiclet, OJSC. LDM-MCp0411100101-Q208 Evolution Development kit allows the developer to quickly learn how to work with multi-cell MCp0411100101-Q208I processor and to control its peripheral devices.

2.2 Description of MCp0411100101-Q208I microprocessor

Multi-cell MCp0411100101-Q208I microprocessor is designed for resolve a wide range of operation tasks and digital signal processing in applications requiring minimal power consumption and high performance, such as:

- industrial automation systems from smart sensors to control systems;
- motor control device;
- universal receivers for navigation GLONASS / GPS / Galileo / COMPASS (China) / IRNSS (India) / QZSS (Japan);
- mobile phones;
- visual 3D;
- automotive Electronics for "intelligent" on-board systems that control the traffic situation and warning drivers of the dangers and traffic jams;
- the automatically recognizing “friend-or-foe” identification system

2.3 Technical features of microprocessor

MCp0411100101-Q208I microprocessor incorporates multicellular processor core - the first processor core with a radically new post-Neumann multicellular architecture developed in Russia. Multicellular processor consists of 4 cells (coherent processing units) combined intellectual commutation environment.

Basic features:

- Cells – 4;
- Processor capacity - 32/64 bits;
- Data memory - 128 Kb (4*4K*64);
- Program memory – 128 Kb (4*4K*64);
- ROM – (in case «1» is absent) for storing executable binary external serial FLASH ROM (FLASH XCF04S XILINX is installed on this board) should be used;
- floating point numbers calculation block (in each cell);

- Clock frequency — 100 MHz;
- Processor throughput — 2,4 Gflops.

Basic characteristics:

- Package – QFP-208;
- Temperature conditions – (-60…+125) ^{o}C;
- Max power consumption: 1,08 W;
- Supply voltage:
  - Core – 1,8 V;
  - Peripherals – 3,3 V.

Peripherals:

- 2 SPI with slave selector (in master mode);
- 4 UART with FIFO на прием/передачу;
- 2 I2C (1 «master» and 1 «slave»);
- I2S;
- Ethernet 10/100 Mb/s;
- USB 1.1 FS (device) controller with external serial interface for I/O devices;
- 7 general-purpose timers;
- 4 I/O ports, (104 pins total);
- 4-channel pulse-duration modulation;
- Watchdog timer.

2.4 Technical description

The main structural external and internal components of LDM-MCp0411100101-Q208 Evolution board are shown on fig.2.

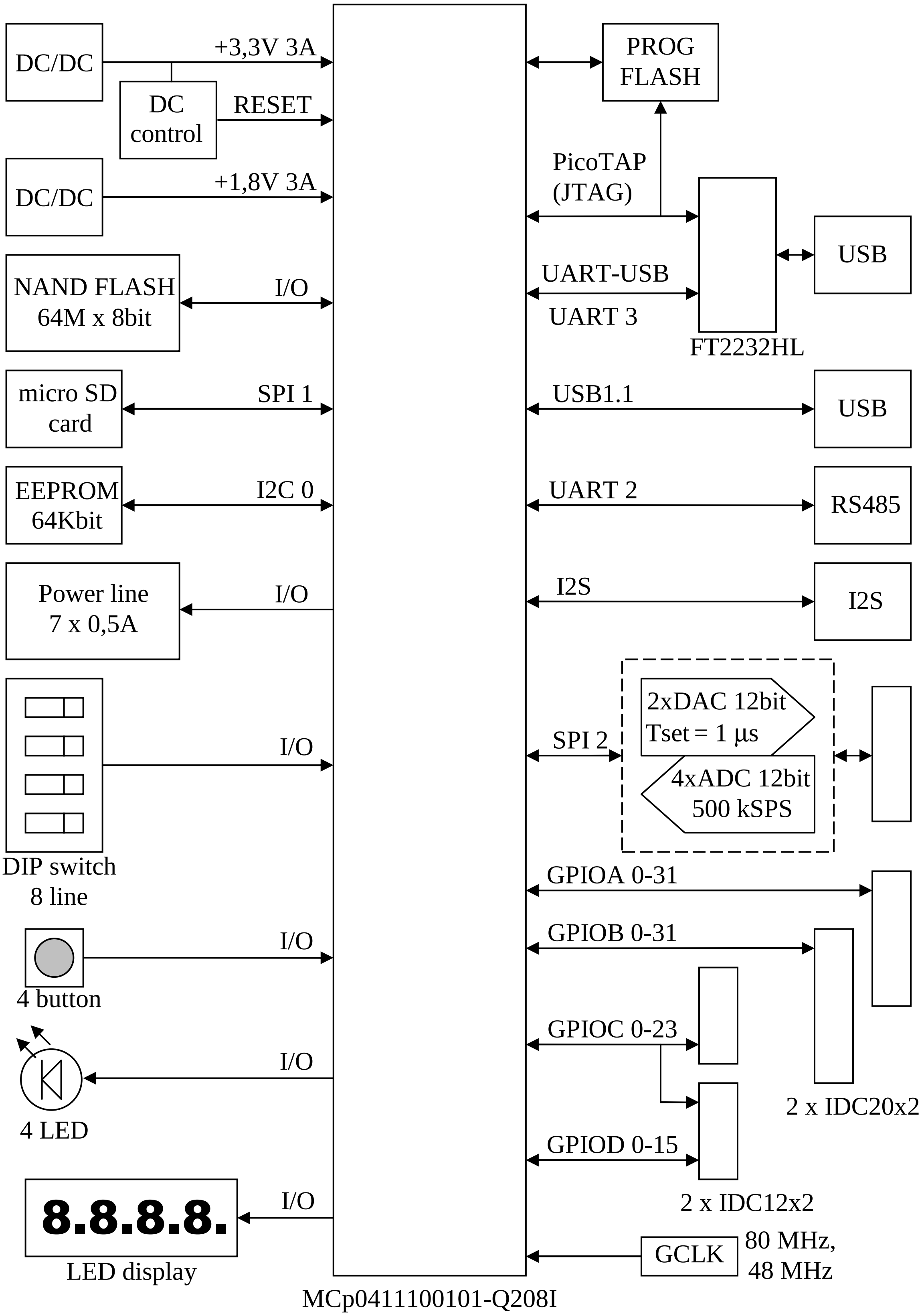


Figure 2. Block diagram of LDM-MCp0411100101-Q208 Evolution board

Components:

- MCp0411100101-Q208I processor;
- clock oscillator 80 MHz (MCp0411100101 core);
- clock oscillator 48 MHz (USB interface);
- DAC – 2 channels, 12 bit, $T_{set} = 1\ \mu s$;
- ADC – 4 channels, 12 бит, 500 kSPS;
- operational amplifier on the output of DAC and on the input of ADC;
- I2C EEPROM 64 Kb;
- NAND FLASH 64 Mb;
- program FLASH (XCF04SVOG20C).

Interfaces:

- USB 1.1 FS, connector Mini USB-A;
- UART-USB, connector Mini USB-A (for PicoTAP);
- I2S, connector PLS-4;
- connector micro SD;
- 7 power supply lines 0,5 A, 30 V;
- UART-RS485, connector TJ4-8P8C.

Debug tools:

- built-in JTAG programmer PicoTAP;
- all ports of microprocessor are divided within 4 connectors;
- control sport on the board.

Controls and indication:

- buttons: «reset», «wake-up», «nmi» and 4 optional buttons;
- LEDs: «MP-ready», «power» and 4 optional LEDs;
- 4 digits seven-segment LED indicator.

Power supply:

- switching power supplies 3,3 V, 3 A and 1,8 V, 3 A on the board;
- when plugging to USB with «Host» mode the board should be powered by 5V (taking into account max current of the USB port);
- maximum current on 5 V line (maximum performance at 80 MHz, without extra communication) – 400 mA;
- power supply of 5-12 V is needed when powering coaxial connector or clip connection;
- all lines have protection from opposition circuit.

Structural specifications:

- board material: FR-4, 1.5 mm, 4 layers with structural and mask;
- dimensions: 140 x 110 x 15 mm.

Delivery set:

- development board LDM-MCp0411100101-Q208 Evolution with MCp0411100101-Q208I processor;
- USB-A - mini USB-B cable;
- CD-R with software and documentation;
- special box container.

2.5 Structure and operation

2.5.1 Operaton controls

Components layout

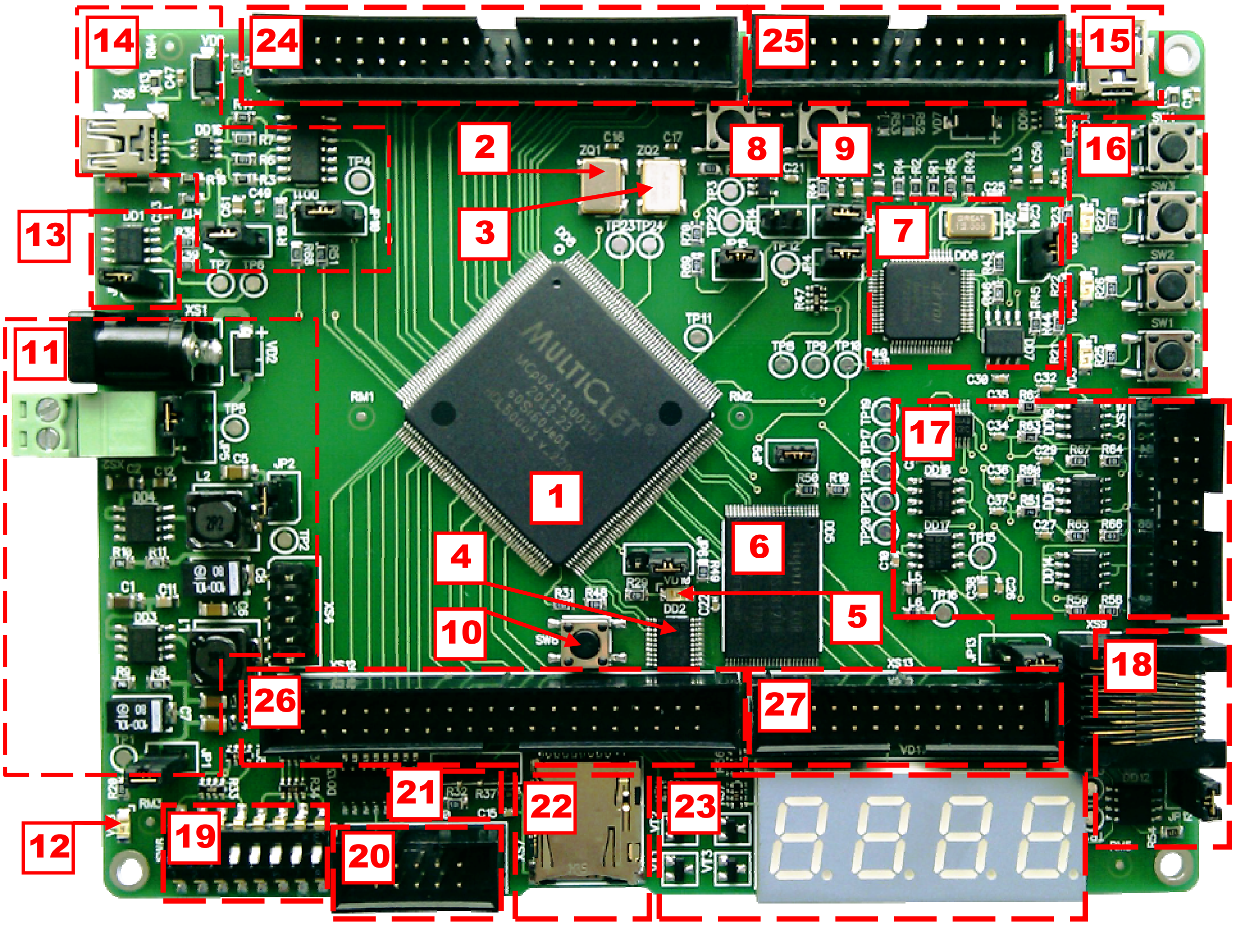


Figure 3. Components layout on LDM-MCp0411100101-Q208 Evolution

##	Description	##	Description
1	Processor MCp0411100101-Q208I	**15**	miniUSB connector, PicoTAP+USB-COM
2	Clock oscillator 80 MHz	**16**	Button and LEDs block
3	Clock oscillator 48 MHz	**17**	ADC, DAC
4	Programm FLASH (XCF04SVOG20C)	**18**	RS485
5	Processor “Ready” indicator	**19**	Switches for 8 I/O
6	NAND FLASH 64 Mb	**20**	Power connector 7x0,5 A, 30 V
7	PicoTAP+USB-COM	**21**	I2S, PLS-4 connector
8	Button «wake-up»	**22**	microSD-card connector
9	Button «reset»	**23**	4 digits LED indicator
10	Button «nmi»	**24**	Port A (32 I/O)
11	DC/DC convertor	**25**	Port D (16 I/O) + port C (4 I/O)
12	Power indicator	**26**	Port B (32 I/O)
13	EEPROM 64 Kb (I2C)	**27**	Port C (20 I/O)
14	USB 1.1 (miniUSB connector)		

Controls description

Jumpers:

##	Position 1-2	Position 2-3	Default
JP1	Connection to power line 3.3 V (powering I/O ports of processor)	-	1-2
JP2	Connection to power line 1.8 V (processor core power)	-	1-2
JP3	Connection of TCK PicoTAP line to TCK line of processor and PROM	-	1-2
JP4	Connection of RST PicoTAP line to JNTRST line of processor	-	1-2
JP5	Board power from external power supply (XS1 and XS2 connectors)	Board power from USB port (XS3 and XS6 connectors)	1-2
JP6	Connection of TMS PicoTAP line to TMS line PROM (DD2) XCF04SVO20C	Connection TMS PicoTAP line to TMS line of processor	1-2
JP7	PicoTAP program mode	USB-COM (when jumper is absent)	1-2
JP8	Connection of SCL line EEPROM SCL line of I2C of processor	-	1-2
JP9	Connection of PC16 processor line to NAND FLASH activation line (if memory is not used -jumper is out)	-	1-2
JP10	Setup mode of USB bridge USB1T11 (MODE = 0)	Setup mode of USB bridge USB1T11 (MODE = 1)	1-2
JP11	Connection of OE' line of USB bridge USB1T11 to processor operation line	-	1-2
JP12	Connection of 120 Ohm shunt resistor to RS485 line	-	1-2
JP13	Connection of 3,3 V power line to 1, 2 pins of XS9 connector (RS485) (if power to external devices from RS485 not needed – jumper is out)	Connection to power lines of XS1, XS2 to 1 and 2 pins of XS9 connector (RS485) (if power to external devices from RS485 not needed – jumper is out)	1-2
JP14	Connection of processor line PC19 to WDI controller (3,3 V)	-	Absent
JP15	Connection of RST PicoTAP line to the line of general RESET	-	1-2

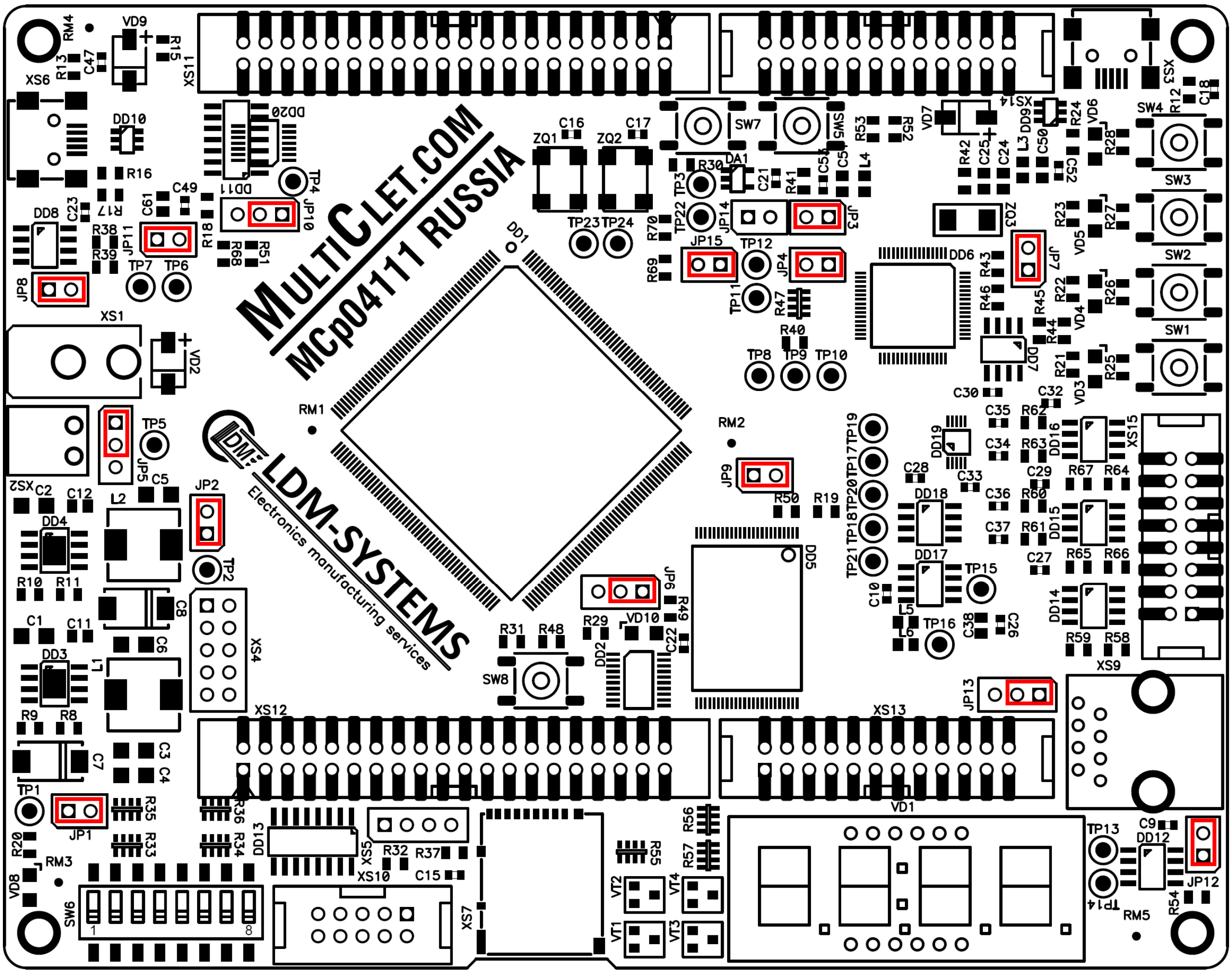


Figure 4. Default jumpers position on LDM-MCp0411100101-Q208 Evolution board

Buttons:

SW1 – SW4 – user buttons;

SW5 – general RESET button;

SW7 – «wake-up» button;

SW8 – nonmaskable interrupt «nmi».

Indication:

VD3 – VD6 – user LEDs;

VD8 – Power LED +3,3 V;

VD10 –MCp memory READY.

2.5.2 Drivers and software setup

Before working with the board it is necessary to install a driver of FT2232 chip of USB-COM interface and JTAG-controller PicoTAP driver. The required drivers are on the CD, included in the delivery, or they can be downloaded from manufactures websites (www.goepel.com, www.ftdichip.com).

STEP 1: The driver of chip FT2232 of USB-COM interface setup

Linux

The latest version of the driver for Linux and instruction for its installment are on websites above. Also you may use the driver, including in the delivery.

On the CD of the Development Kit find the file

CD:\DRV\FT2232\Linux\libftd2xx1.1.0.tar.gz:

1. Open the file into the appropriate folder on the disk:
   - gunzip libftd2xx1.1.0.tar.gz;
   - tar -xvf libftd2xx1.1.0.tar.
2. Change the folder on the required for architecture MP of your PC: build/i386 – for 32-bit system or build/x86_64 - for 64-bit system.
3. Under the user name «root» copy files mentioned below into the folder /usr/local/lib:
   - cp libftd2xx.so.1.1.0 /usr/local/lib.
4. Change the folder to /usr/local/lib:
   - cd /usr/local/lib.
5. Create a link to files:
   - ln -s libftd2xx.so.1.1.0 libftd2xx.so.
6. Change the folder to /usr/lib:
   - cd/usr/lib.
7. Create a link to the driver:
   - ln -s /usr/local/lib/libftd2xx.so.1.1.0 libftd2xx.so.

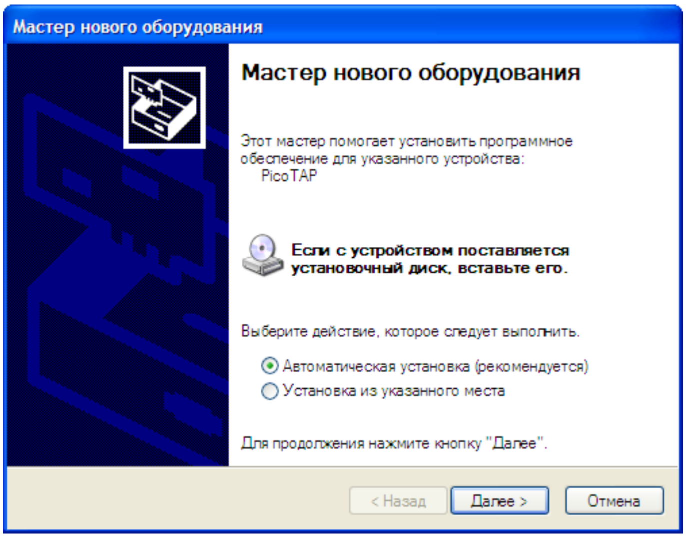
Windows

1. On the CD of the Development Kit find and execute the file CD:\DRV\FT2232\Windows\CDM20824_Setup.exe, follow setup procedure.

STEP 2: JTAG-controller PicoTAP driver setup

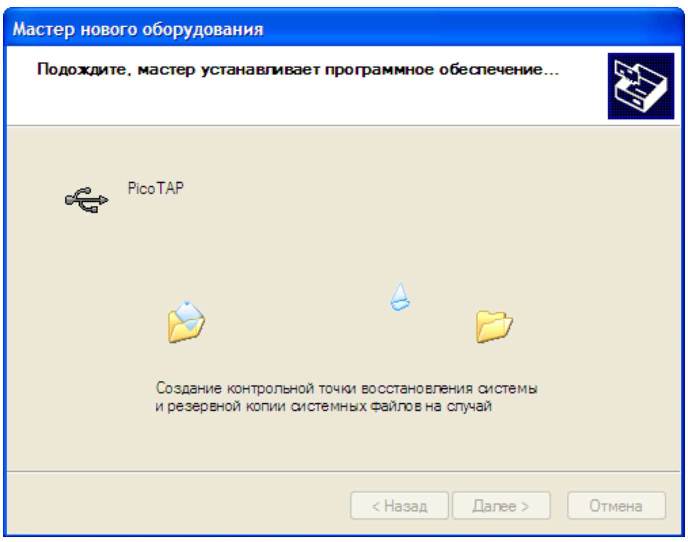
1. On the CD of the Development Kit find and execute the file CD:\DRV\PicoTAP\Pico_TAP_ohnegoJtag(DRV).exe, follow setup procedure.

2. Plug LDM-MCp0411100101-Q208 Evolution board to PC, the following window appears:

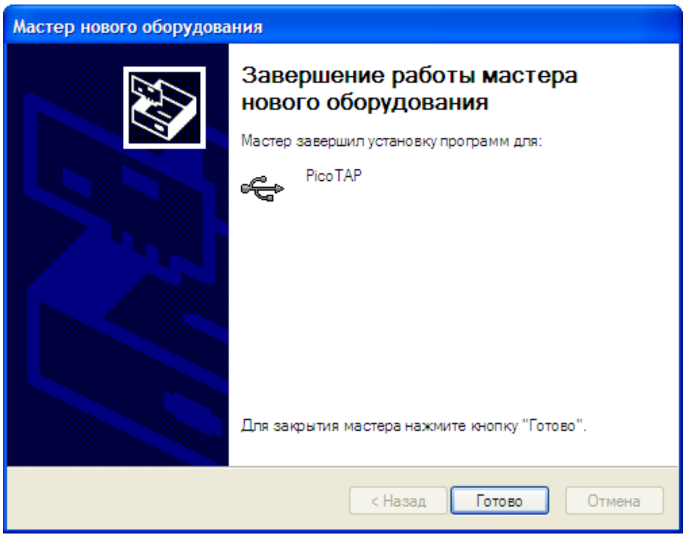


3. Choose automatic installation software procedure, press ”Next”.

4. If needed confirm driver setup for not-tested device PicoTAP, press ”Continue Anyway”.



5. After installing drivers press “Finish” in the final window:



6. The system again informs you about the found device PicoTAP. It is normal because bridge FT2232 consists of two ports PicoTAP. Repeat actions from par.3,

7. After finishing installation the system informs you that the device is connected and works properly.

STEP 3: Compiler Installation

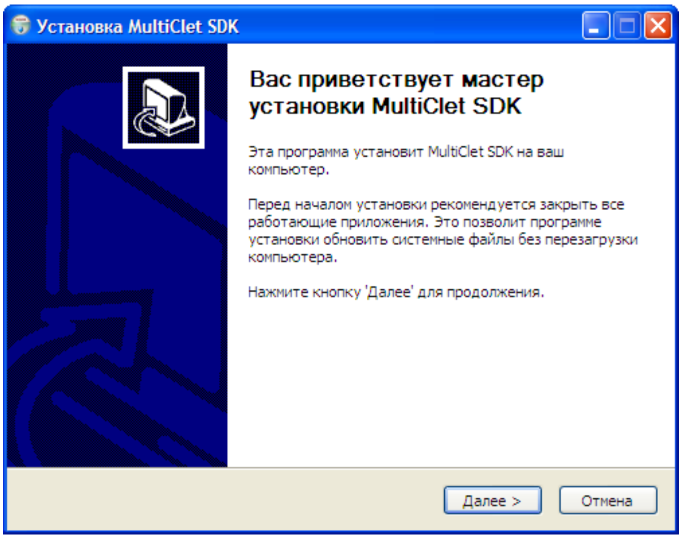
Compiler distribution is on the disk included in the delivery, also you may download it from the website of manufacturers (www.multiclet.com).

1. On the disk included in the delivery find files and install its:

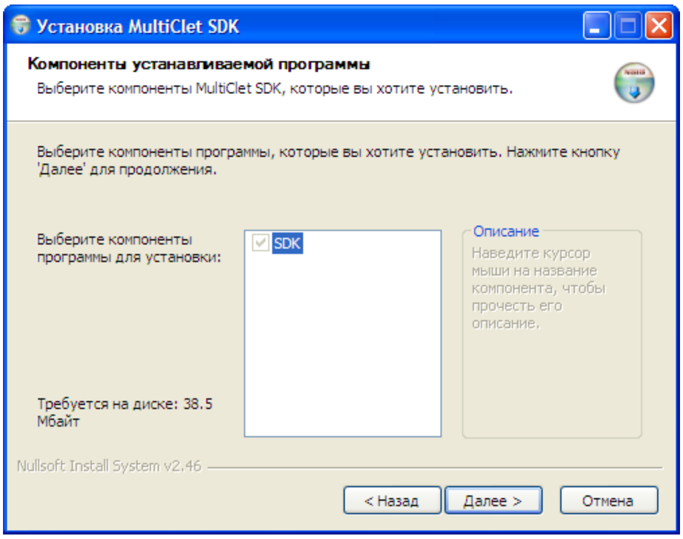
- Linux: CD:\Compiler\Linux\MultiCletSDK.20121205.tar.gz;
- Windows: CD:\Compiler\Windows\MultiCletSDK.20121205.exe.

For example of OC Windows:

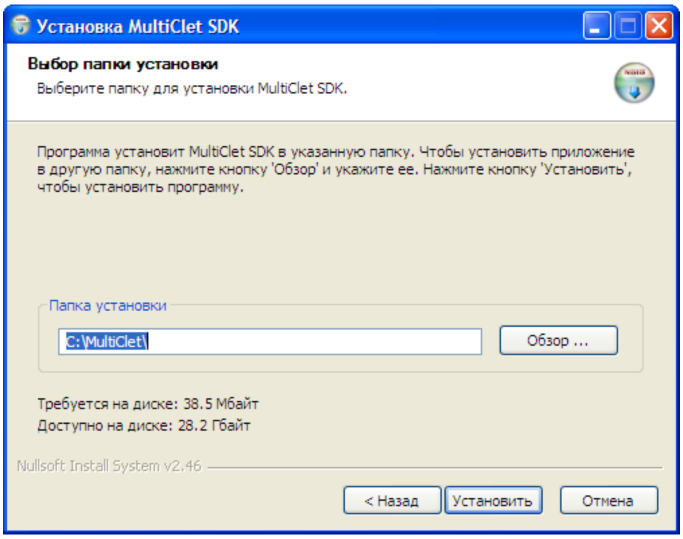
2. In the welcome screen press «Next»



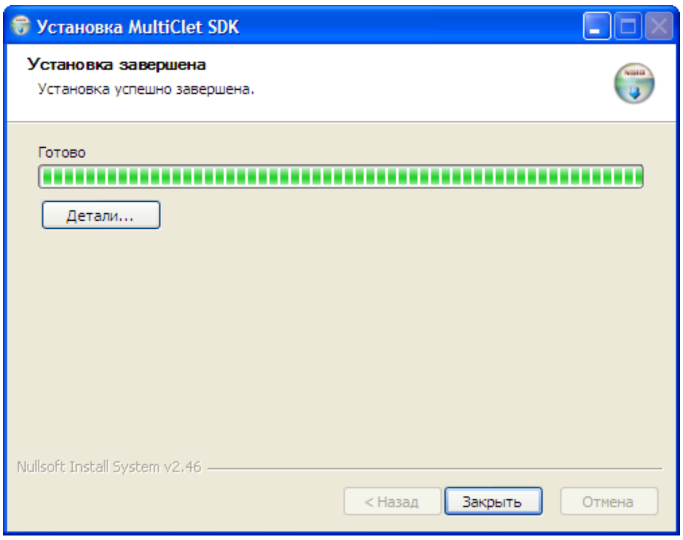
3. In the window «Installing program components» press «Next»



4. In the window «Select setup folder» leave the default and press «Install»



5. In the window «Complete setup» press «Finish»



6. Compilator has been installed.

2.5.3 Device application

Preparing for use, connection

1. Remove the card, USB cable and a CD with compiler and drivers from the box.

2. Install the drivers for the JTAG-Controller (PicoTAP) and for an interface chip USB-UART (FT2232) (§ 2.5.2).

3. Check the jumper settings (Section 2.5.1, Fig. 4). If you do not have the power supply 9-12 V, the board for testing can be powered by any USB port (XS3 or XS6). To enable this feature, you need to put the jumper JMP5 from 1-2 to 2-3 position.

4. Connect the USB interface cable to the connector XS3. LED Power supply VD8 is activated, operating system detects a PicoTAP device.

Now the board is ready for operation.

Verifying functionality of the board

Once the device is ready for operation, check the following:

- VD8 and VD10 LEDs should be ON, (VD10 getting ON later than VD8). LED display VD11 will show the number “0” and it will increase by 1 every second. (preloaded software in the processor on the run);
- Use a voltmeter to measure the voltage test points: TP5 - +5 V ± 5% (or 9-12V), TP1 - +3.3 V ± 5%, TP2 - +1.8 V ± 5%;
- Press and release the «RESET» button, (VD10 and VD11 must go out). Once the button was released, after a while VD10 should light (less than 1 sec), and VD11 will display again 0,1,2,3… .

If any other behavior, please contact the card manufacturer.

Creation test program

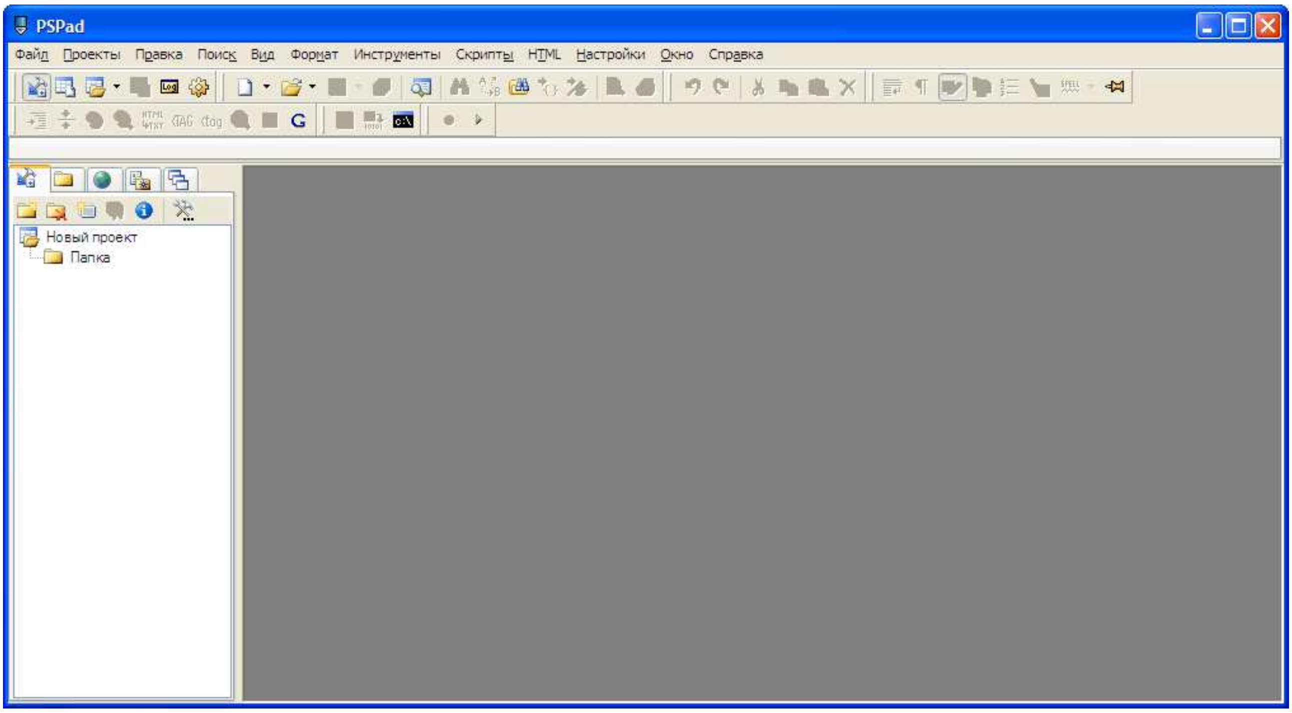
To quickly learn how to work with LDM-MCp0411100101-Q208 Evolution, we present a step by step example of creating a simple code.

STEP 1: Description of the software algorithm

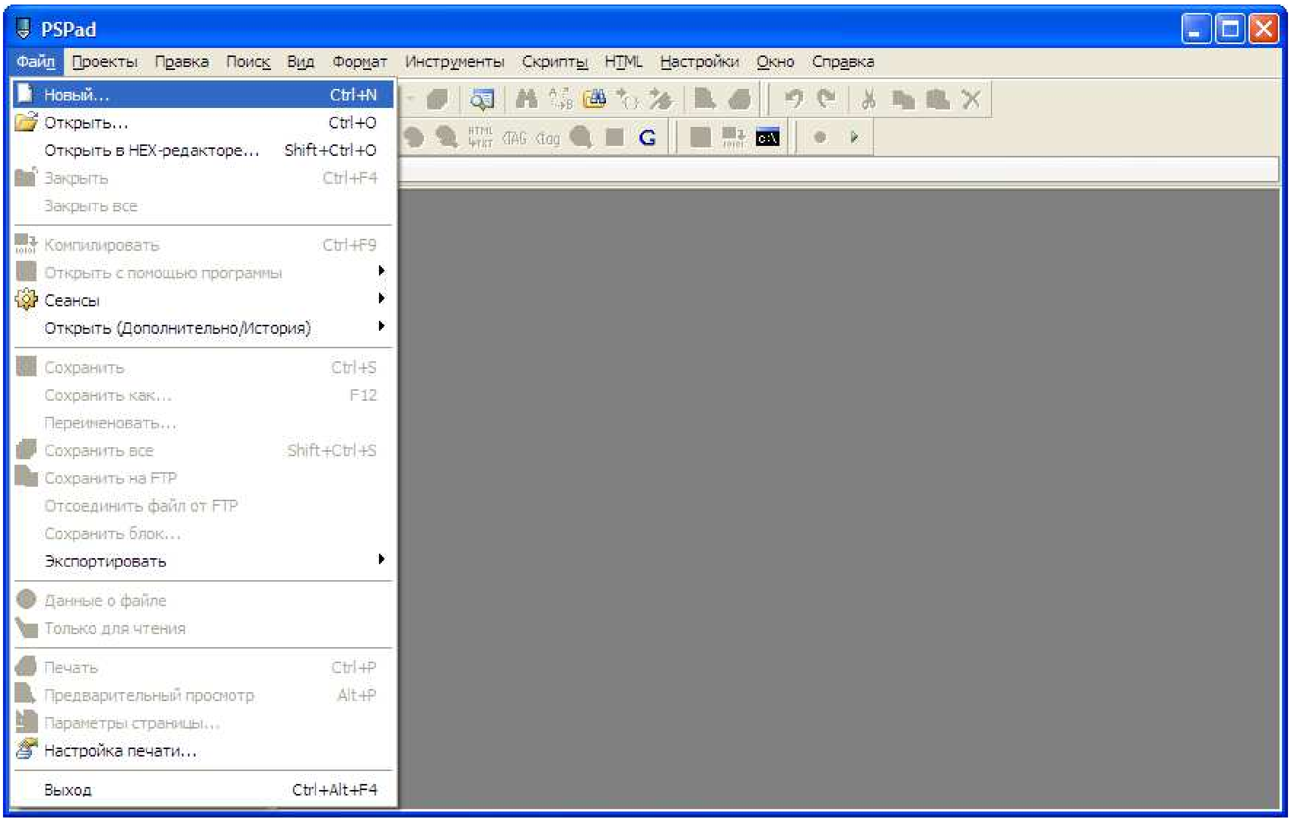
VD11 LED displays numbers from 0 to 9. Increasing the value of the number in the display is produced when you press the SW2. Pressing the button SW1 decreases the number in the display.

STEP 2: Create a new project in the MultiCletSDK

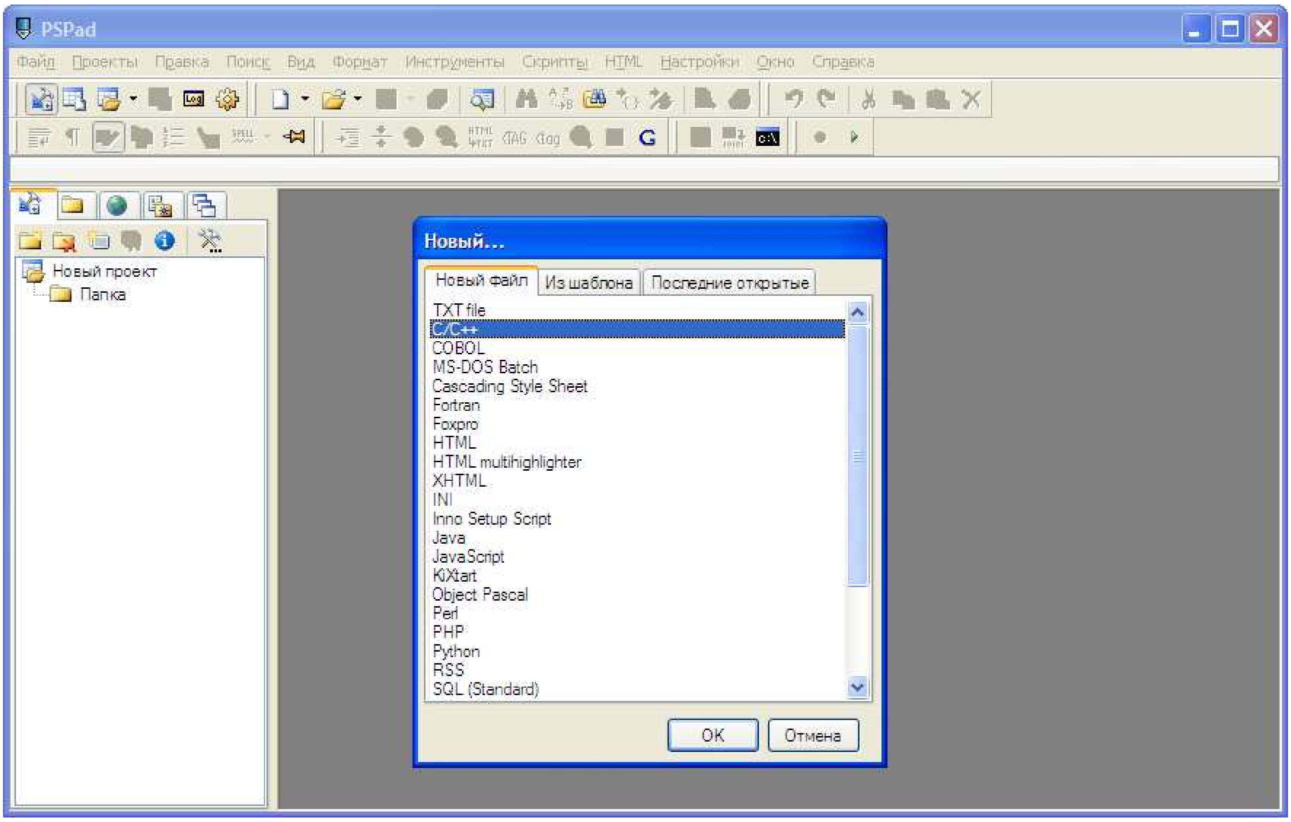
1) Go to “Start/All programms/MultiCletSDK/ PSPad. PSPad main window will appear:



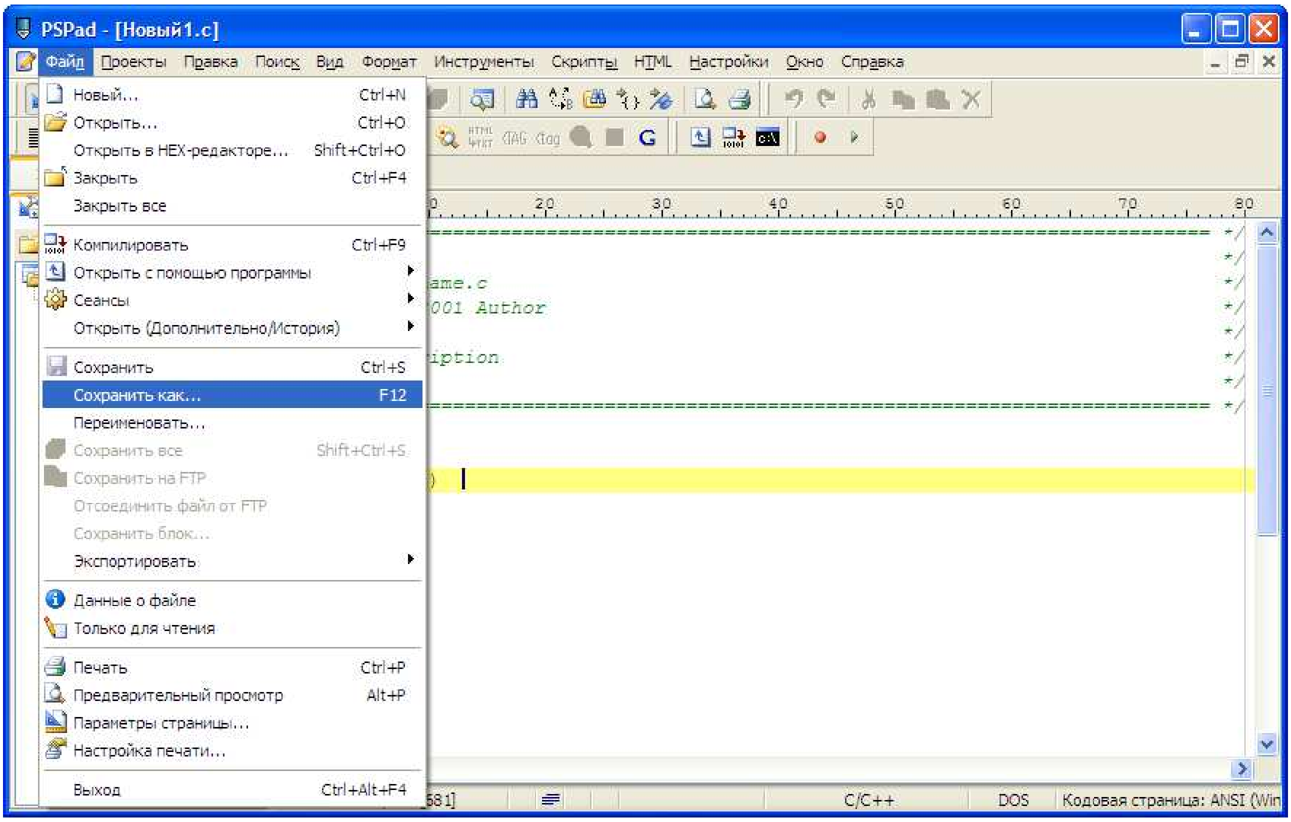
2) Open in «File» menu - «New».

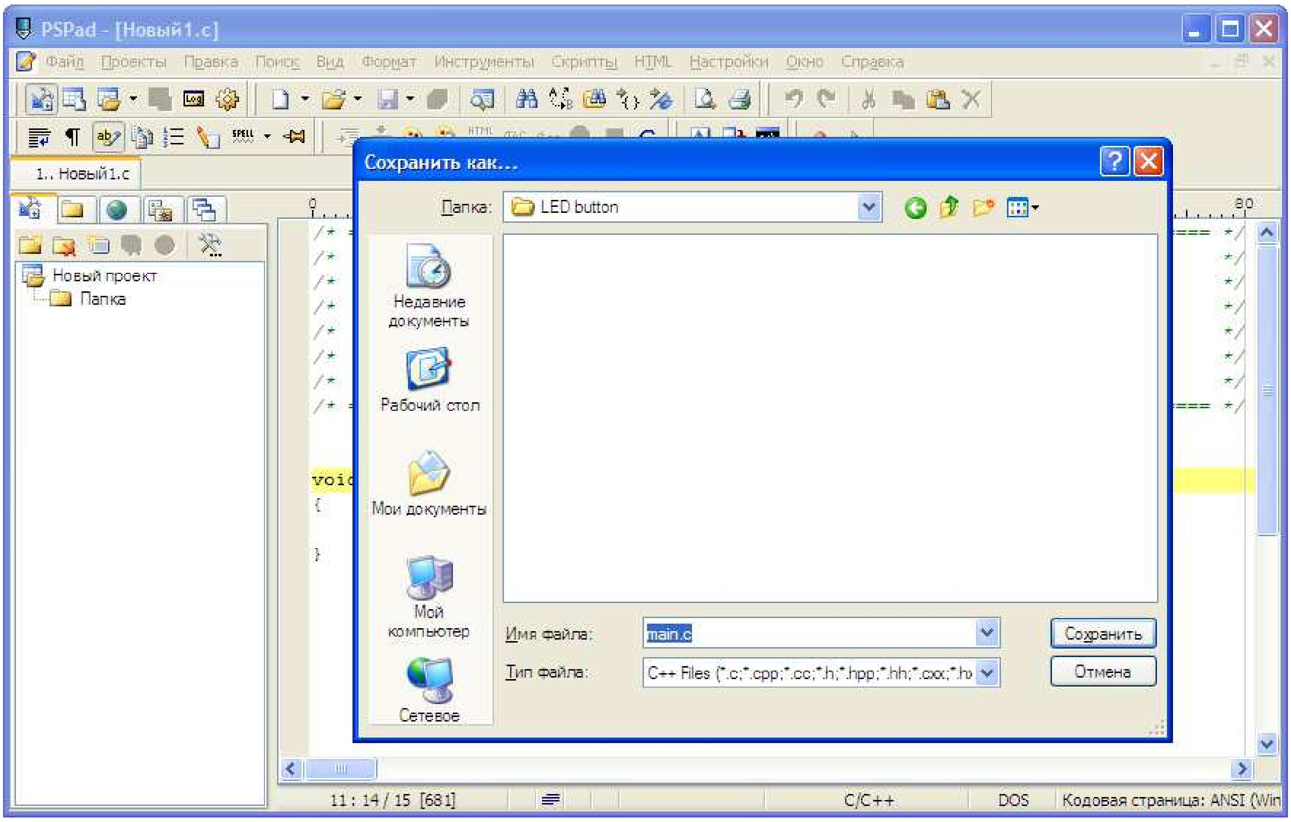


3) In the next window select C/C++ and press OK.

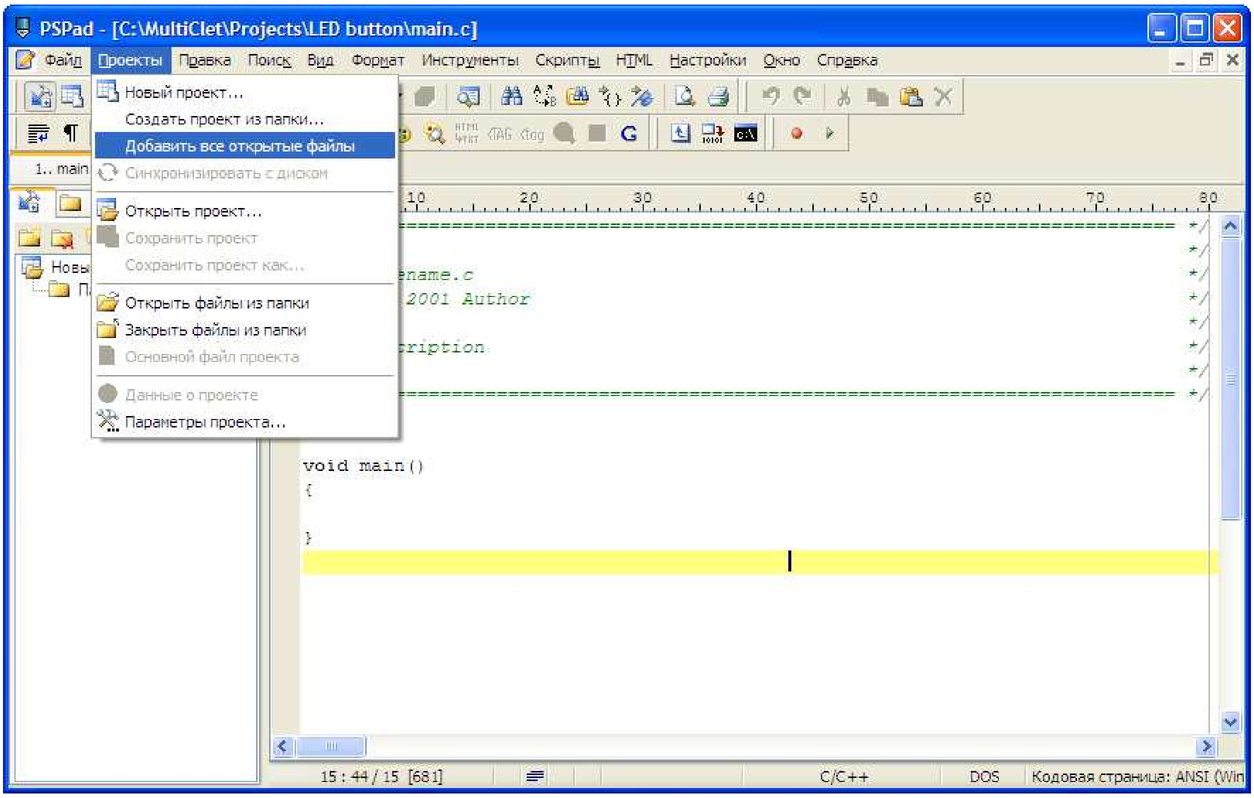


4) Save created file. Select «File» menu then select «Save as…». Create a new subfolder «LED button» in «C:\MultiClet\Projects». Set the filename «main.c» and then press «Save».

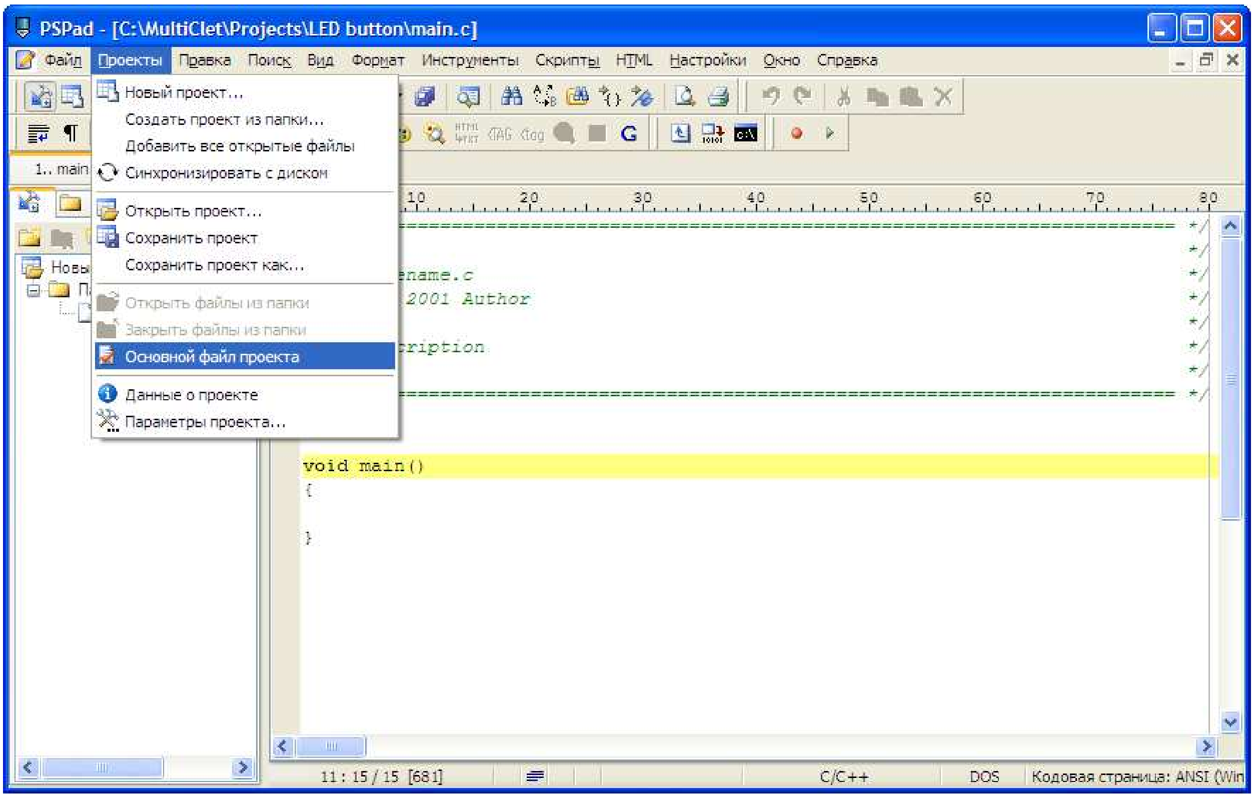




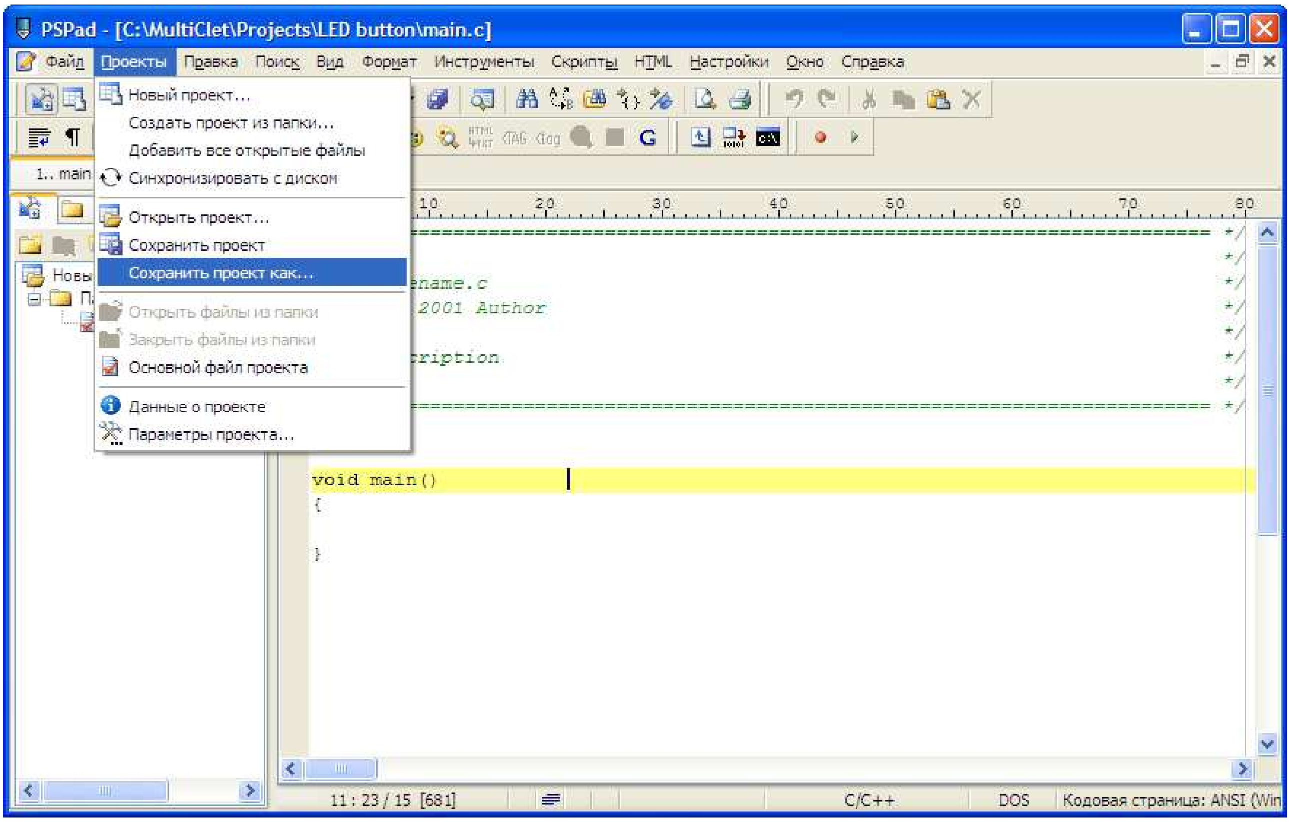
5) Add the created file to the project. Select «Projects/Add all open files». After adding a file «main.c» appears in the project tree.

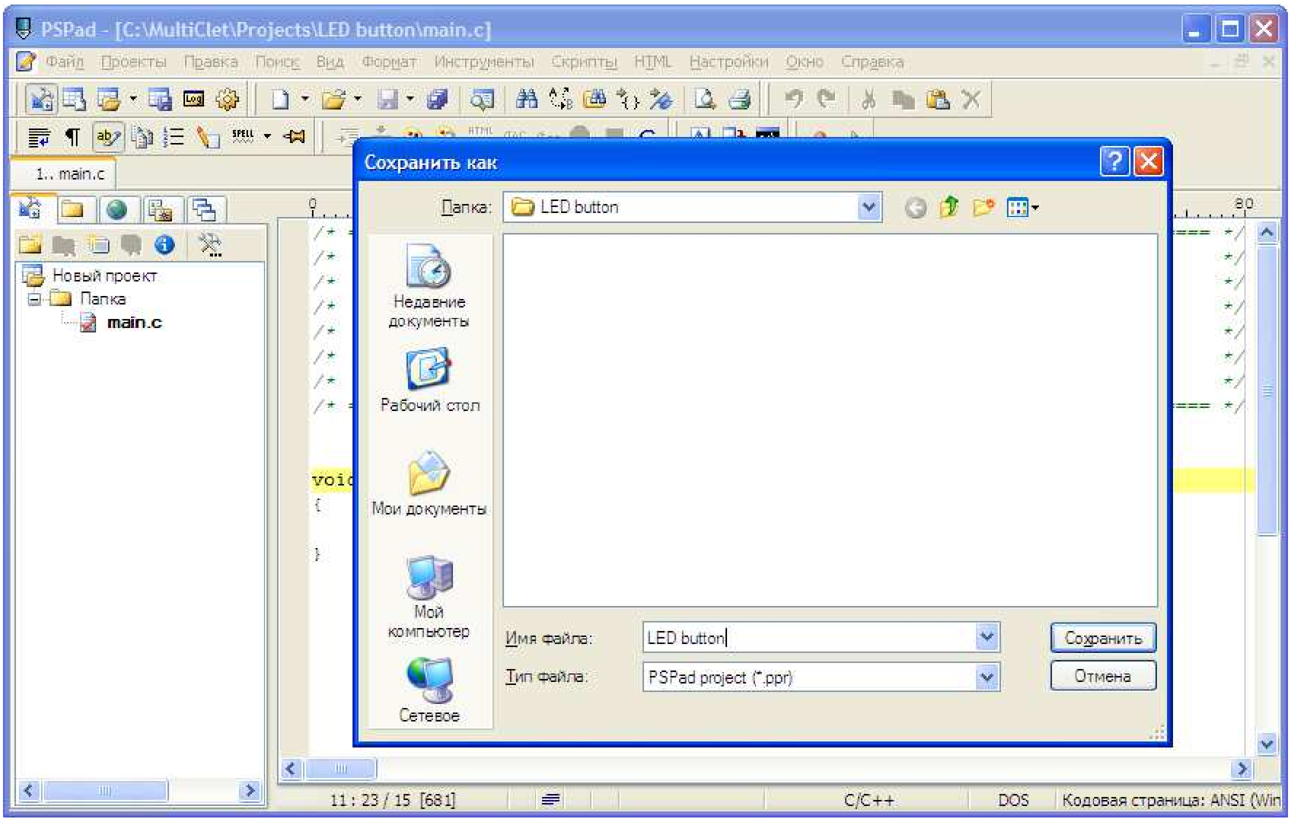


6) Set file "main.c" to be the main project file. Highlight the file main.c in the project tree and follow the "Projects/main file of the project."



7) Save the project. Select «Projects/Save project as…». Set the project name to be «LED button» and press «Save».





Now the new project is created. Tow files main.c and LED button.ppr are in the «LED button» filder.

STEP 3: Create definition and structure of a test program

1) definitions

```
#define    __I    volatile const        /* read-only */
#define    __O    volatile              /* write-only */
#define    __IO   volatile              /* read / write */
```

2) uint32_t variable type

```
typedef unsigned int uint32_t;
```

3) Structure and definitions of a watchdog timer 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) Structure and definitions of B, C and D ports

```
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) Create definitions for easy use of B and D ports

```
#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) Create definitions for LED display lines

```
#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) Create a delay function

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

8) Create a function for LED digits

```
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;
  }
}
```

STEP 4: Correct main() {} function

1) Add variables

```
int status_in;
unsigned char Led_dat = 0;
```

2) Turn WDT off

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

3) Set ports pins for input and output

```
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) Turn on the first digit of a LED display

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

5) Set the start value

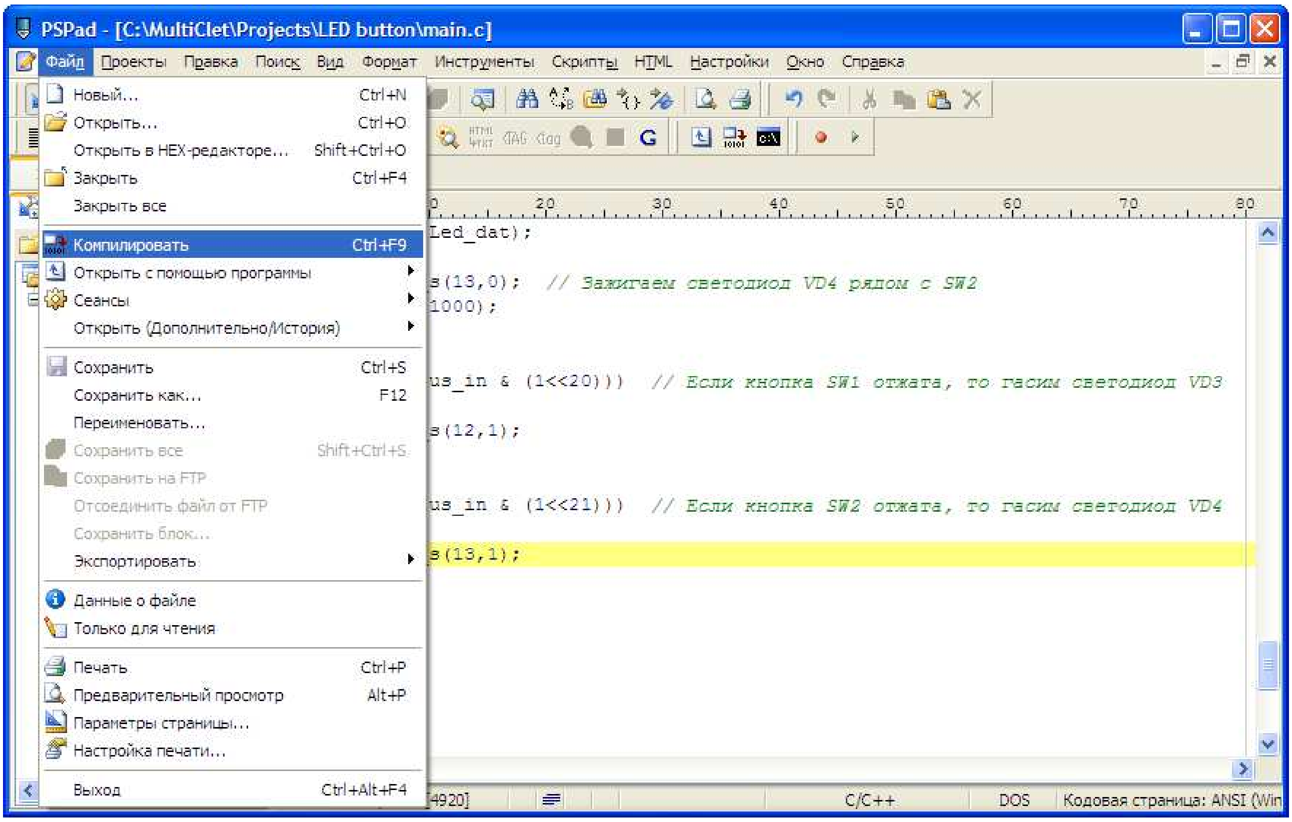
```
DataL(Led_dat);
```

6) Add to the main block

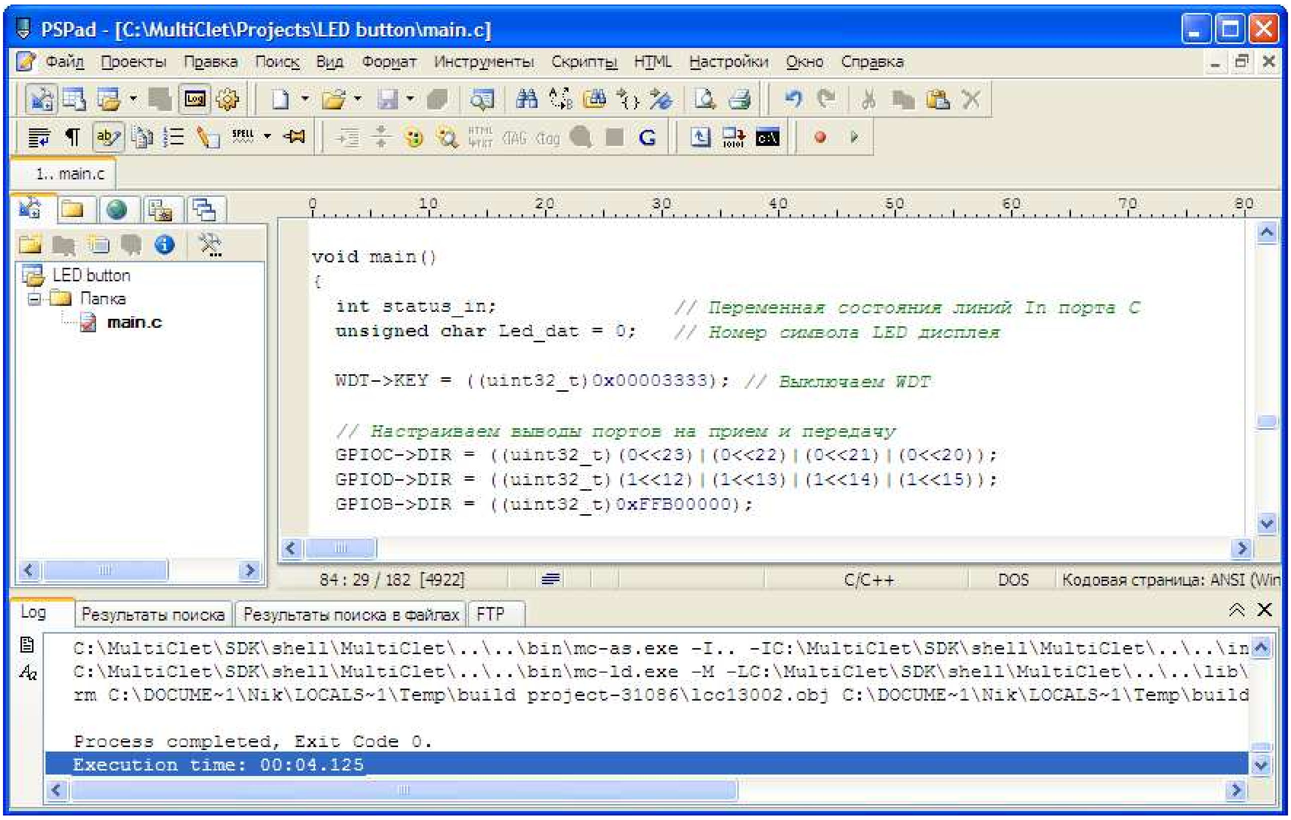
```
while(1)
 {
   status_in = GPIOC->IN;    // Remember the value of the register IN port C

   if(!(status_in & (1<<20)))  // If you press SW1, then decrease the value for LED
   {
     if(Led_dat > 0)
       Led_dat--;
     else
       Led_dat = 9;

     DataL(Led_dat);

     GPIOD_s(12,0);   // Turn on the LED VD3 near SW1
     Delay(1000);
   }
   else if(!(status_in & (1<<21)))   // If you press SW1, then decrease the value for LED
  {
     if(Led_dat < 9)
       Led_dat++;
     else
       Led_dat = 0;

     DataL(Led_dat);

     GPIOD_s(13,0);   // Turn on the LED VD4 near SW2
     Delay(1000);
   }

   if((status_in & (1<<20)))   // If the button SW1 is not pressed, the LED will turn off VD3
   {
     GPIOD_s(12,1);
   }

   if((status_in & (1<<21)))  // If the button SW2 is not pressed, the LED will turn off VD4
   {
     GPIOD_s(13,1);
   }
 }
```

STEP 5: Project compilation

Select «File/compile» or press Ctrl+F9.



Compilation log will be shown at the bottom of the window.



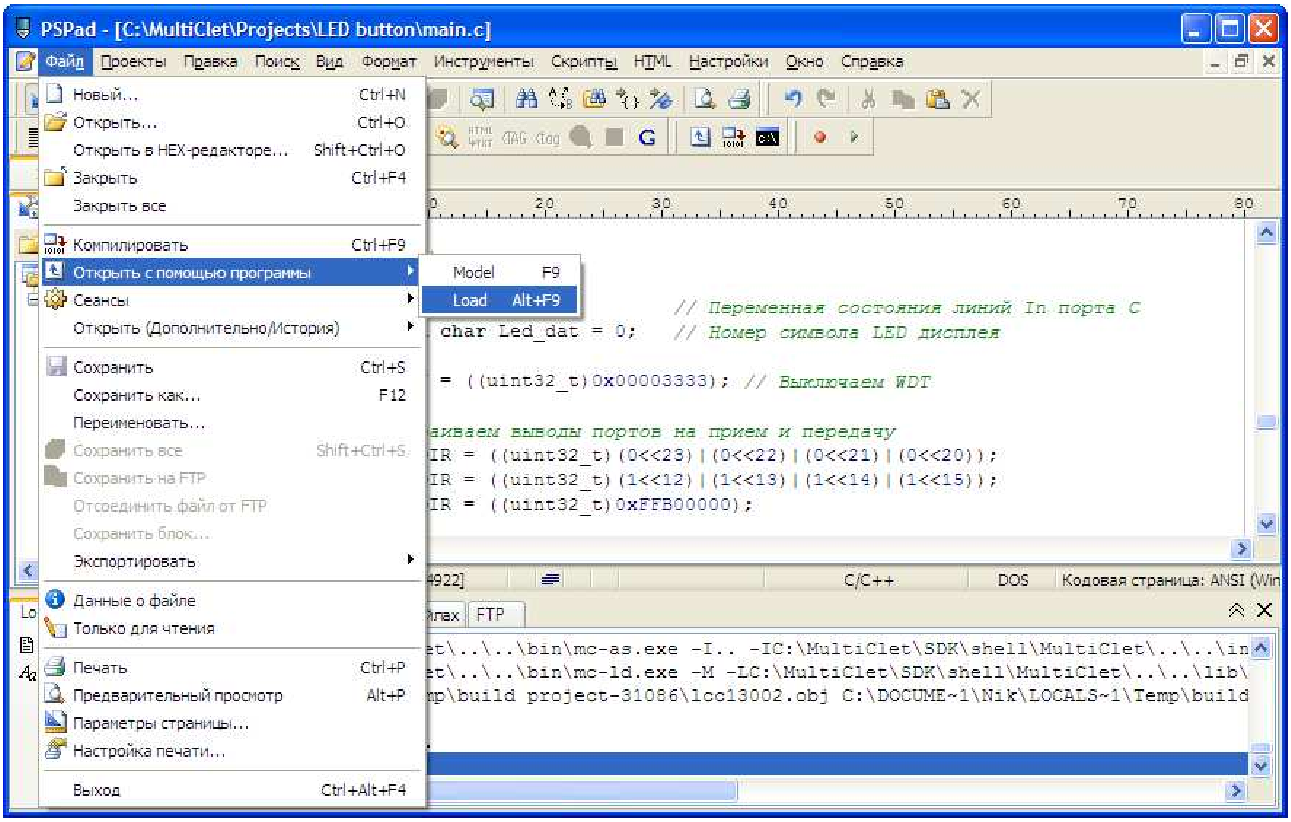
Wait for compilation to be completed:

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

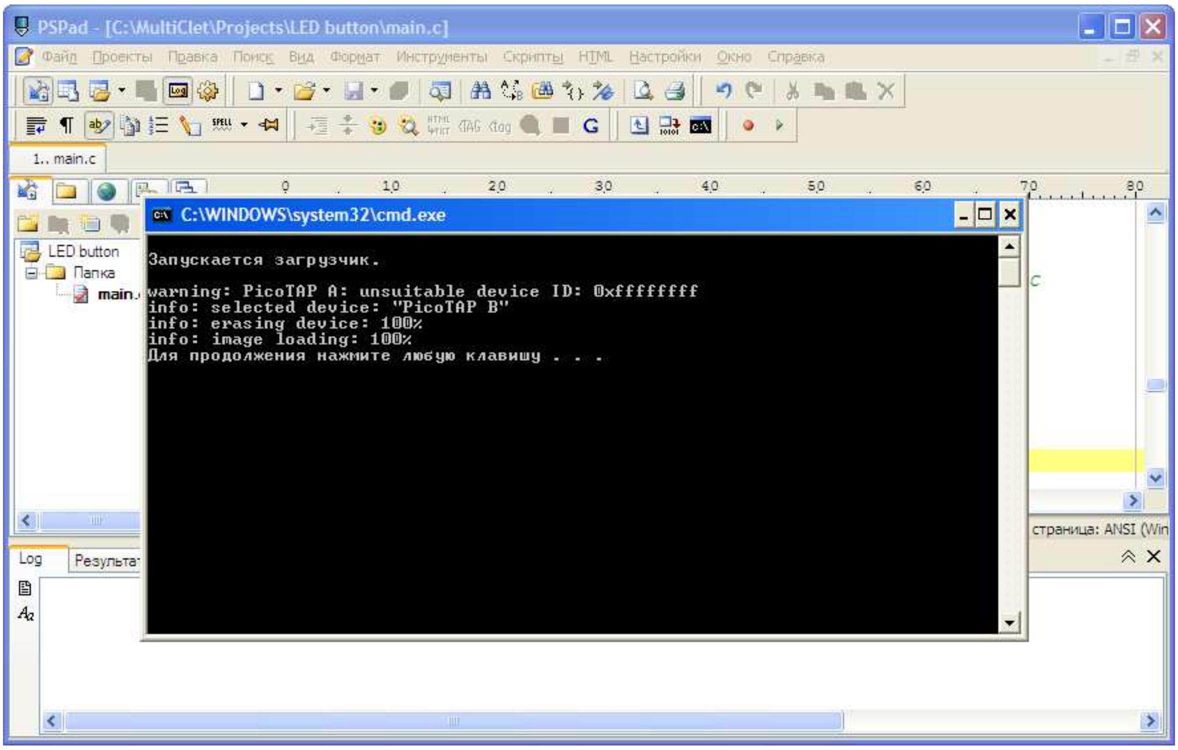
In the project directory folder «out» is created in which a binary file "image.bin" will be placed

Downloading the project to the board

Check the connection of the board to the USB port of a PC and the availability of power by LED VD8. Start the flash process «File/Open with/Load» or press Alt+F9.



An appeared window will show process of erasing the memory and loading the project:



After loading is completed press RESET button (SW5). If loading is OK – LED VD10 is ON. Now you can test the functionality of the program.

LED display will show «0».

Make a single press of a button SW2. Value on the display will change to "1". The next presses will increase this value up to "9". If you press the button SW2 again the display will show "0", and the algorithm will operate in the same cycle.

If pressing the button SW1, number on the display will decrease.

If you have mastered this section, you can now create your own algorithm and test a variety of interfaces: UART, USB, I2S, I2C, SPI, and other peripheral units of the processor and the board: ADC, DAC, EEPROM, etc.

2.6 Electrical scheme of development kit

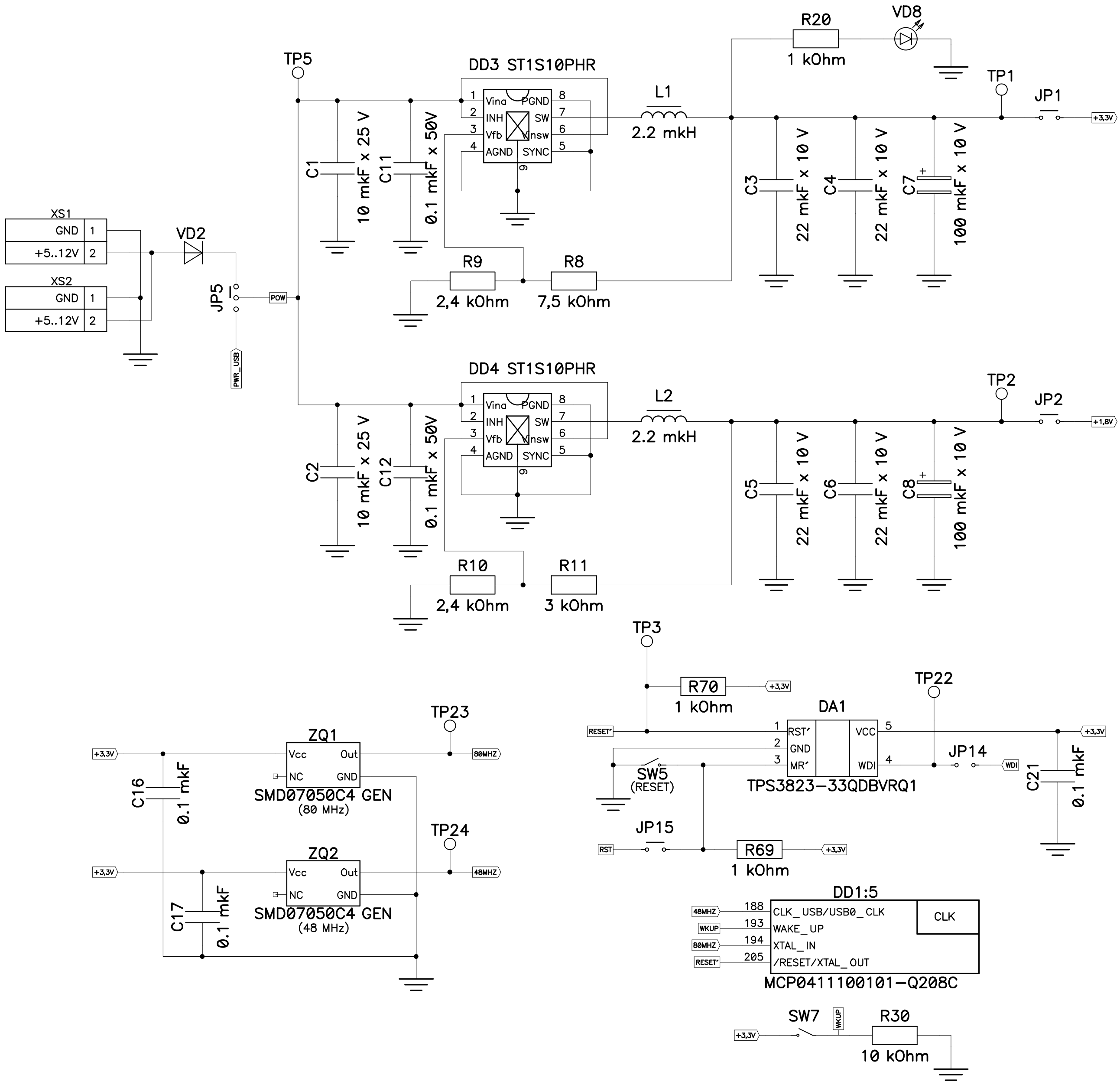


Figure 3. Electrical scheme of voltage converters 3,3V and 1,8 V

For easy control of electrical characteristics there are contact spots on the board (TPx) for basic lines.

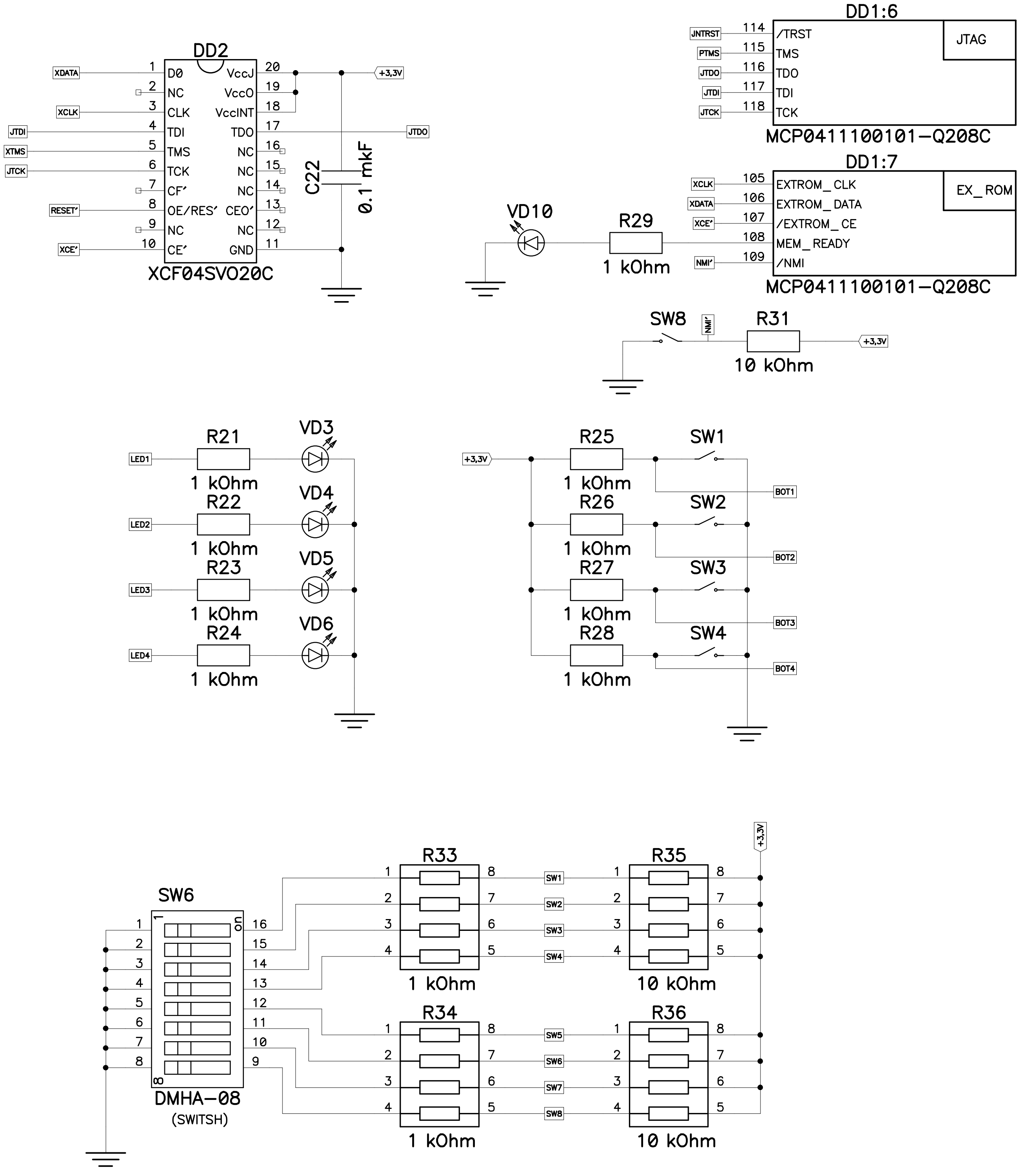


Figure 4. Electrical scheme of FLASH ROM, program processor moule, ports I/O

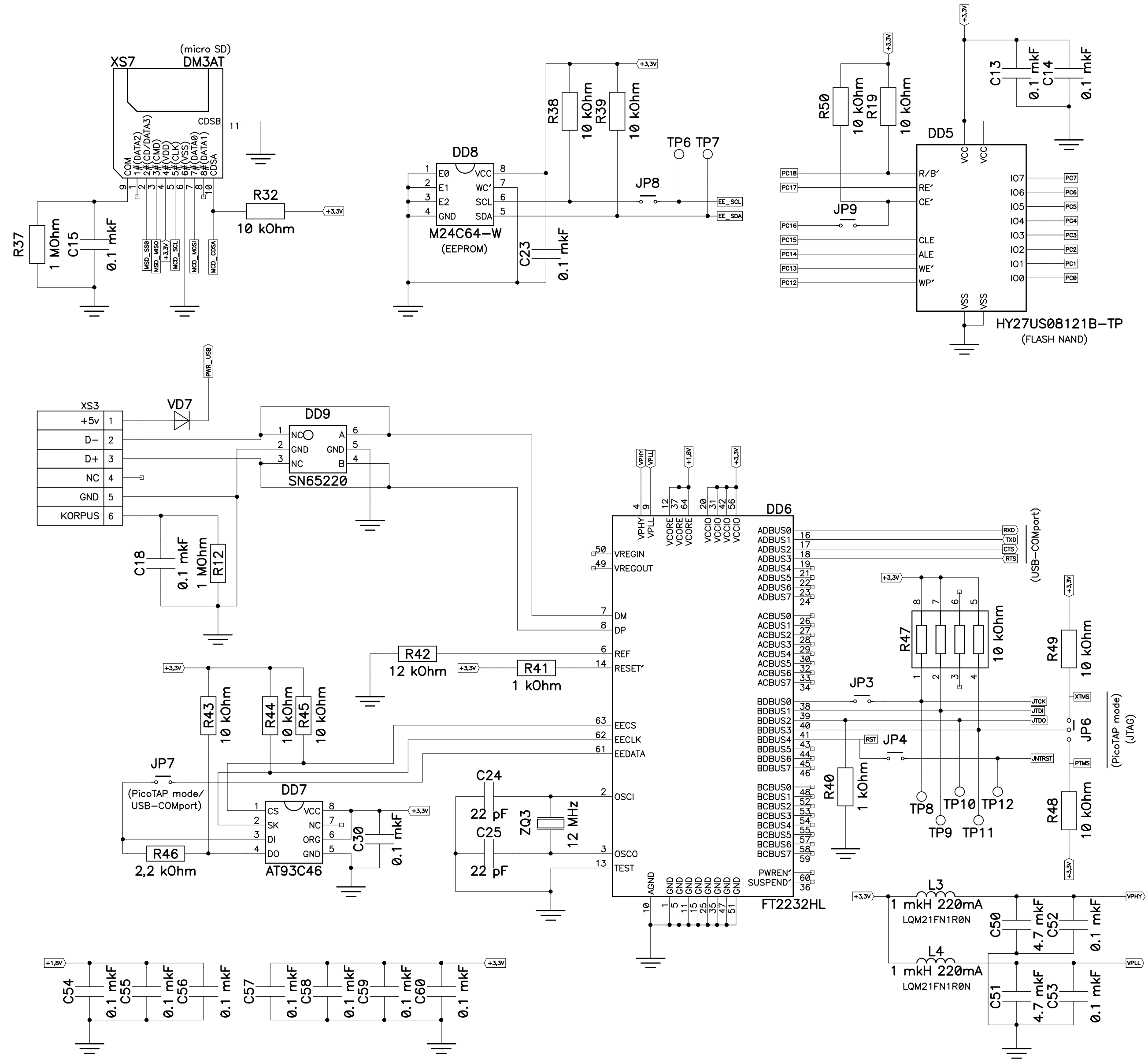


Figure 5. Electrical scheme of microSD card, EEPROM, NAND FLASH, PicoTAP programmer + USB-COM port

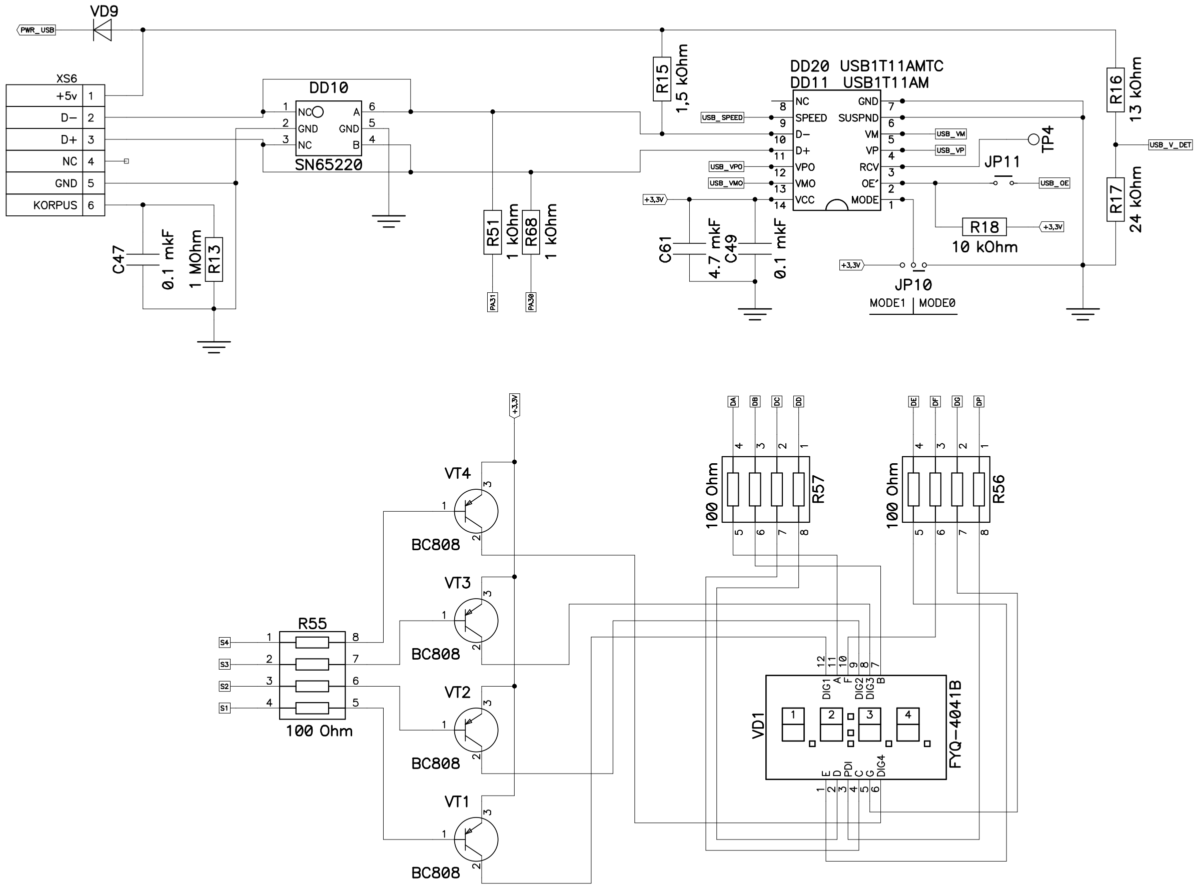


Figure 6. Electrical scheme of USB port, 4 digits LED display

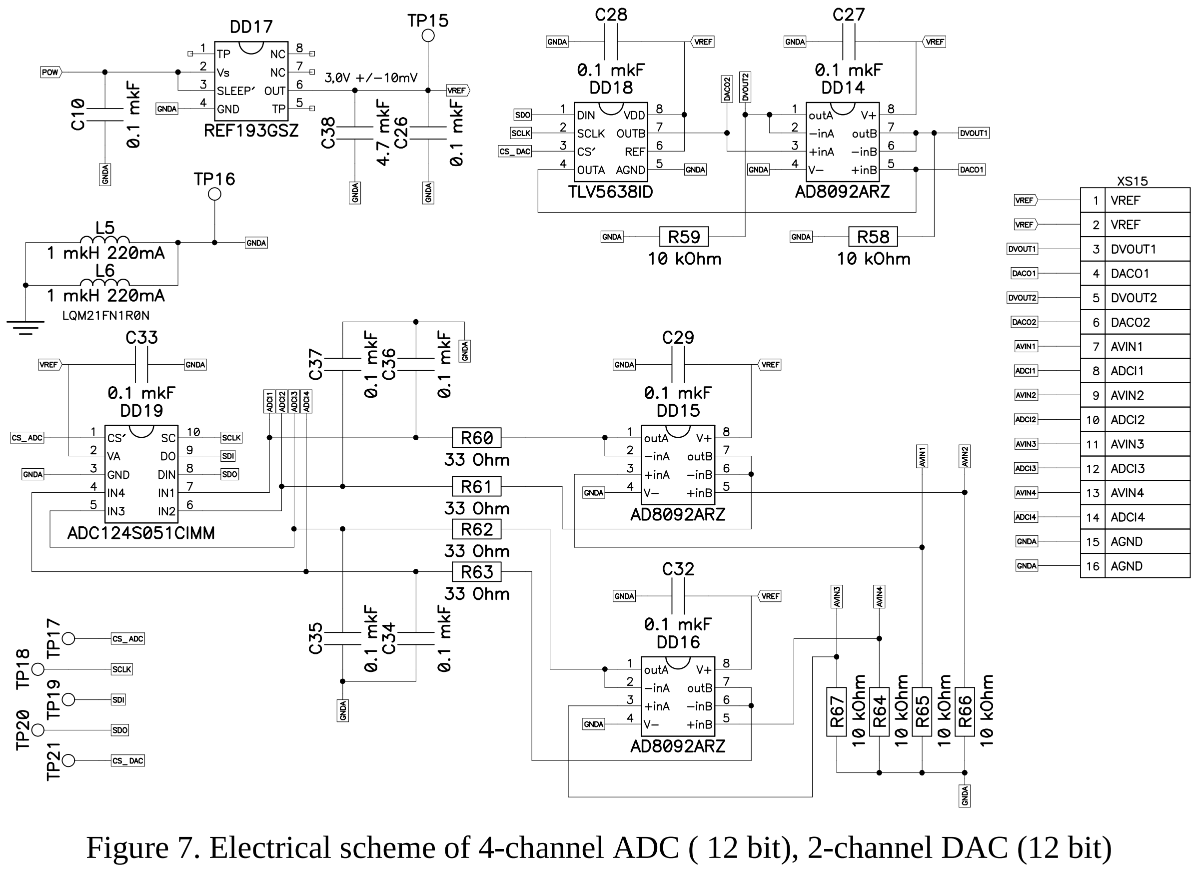


Figure 7. Electrical scheme of 4-channel ADC (12 bit), 2-channel DAC (12 bit)

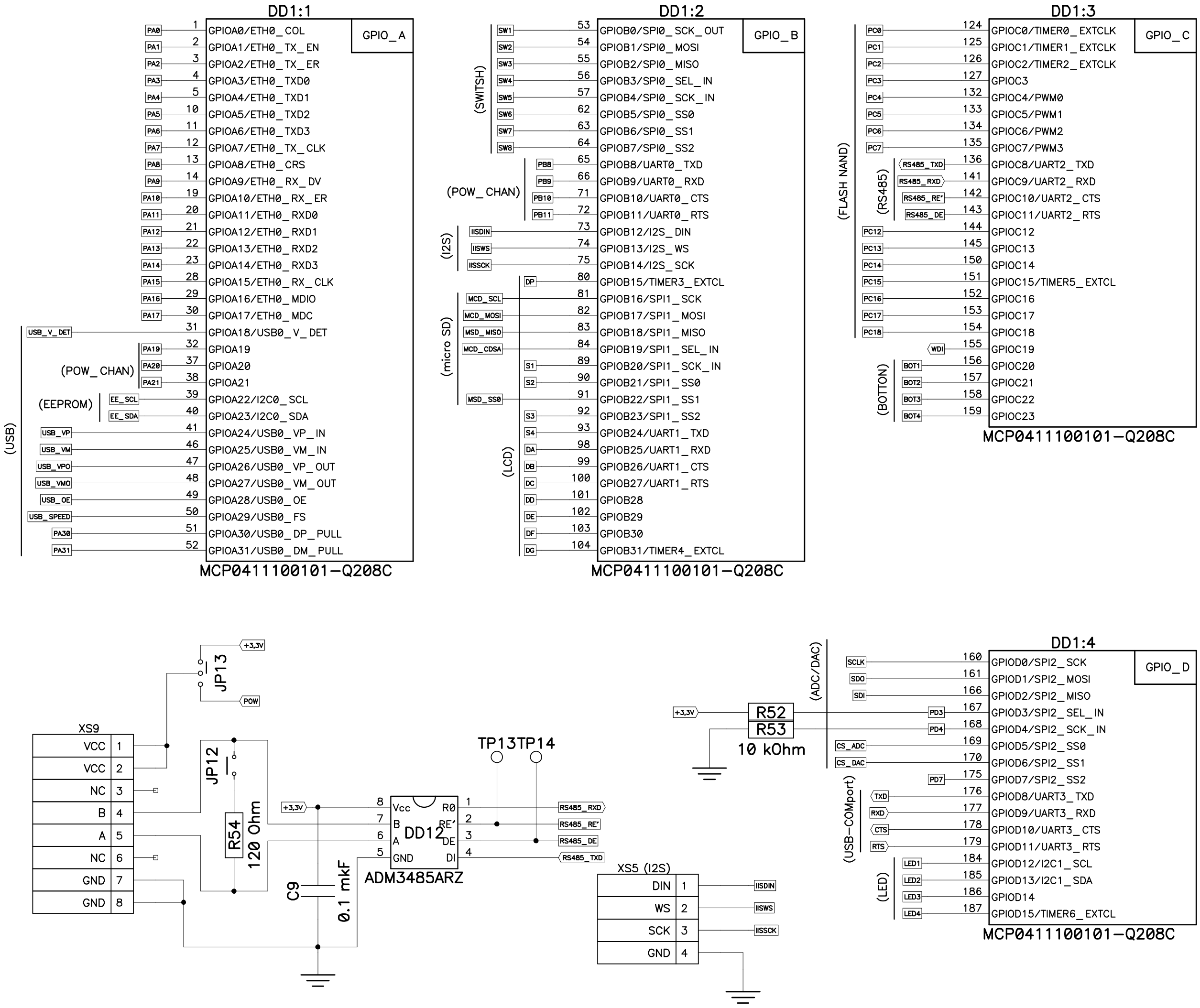


Figure 8. Electrical scheme of processor I/O, RS485 and I2S

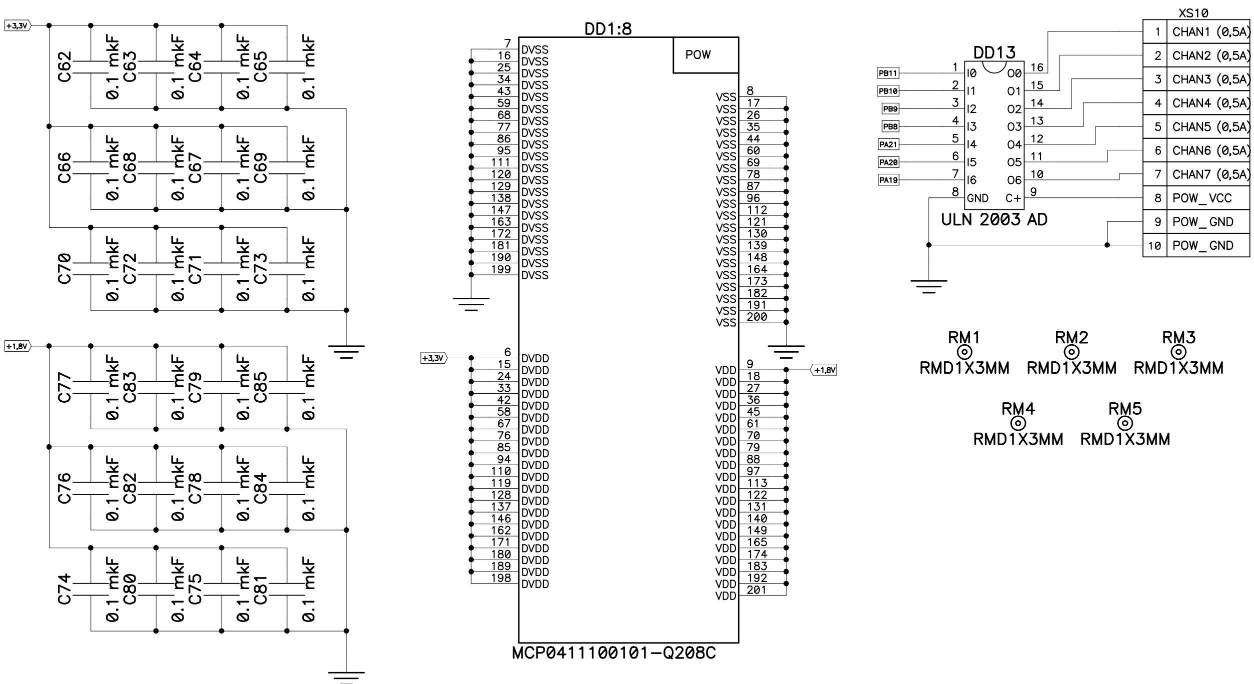


Figure 9. Electrical scheme of processor processor powering (7 channels 0,5 A, 30 V)

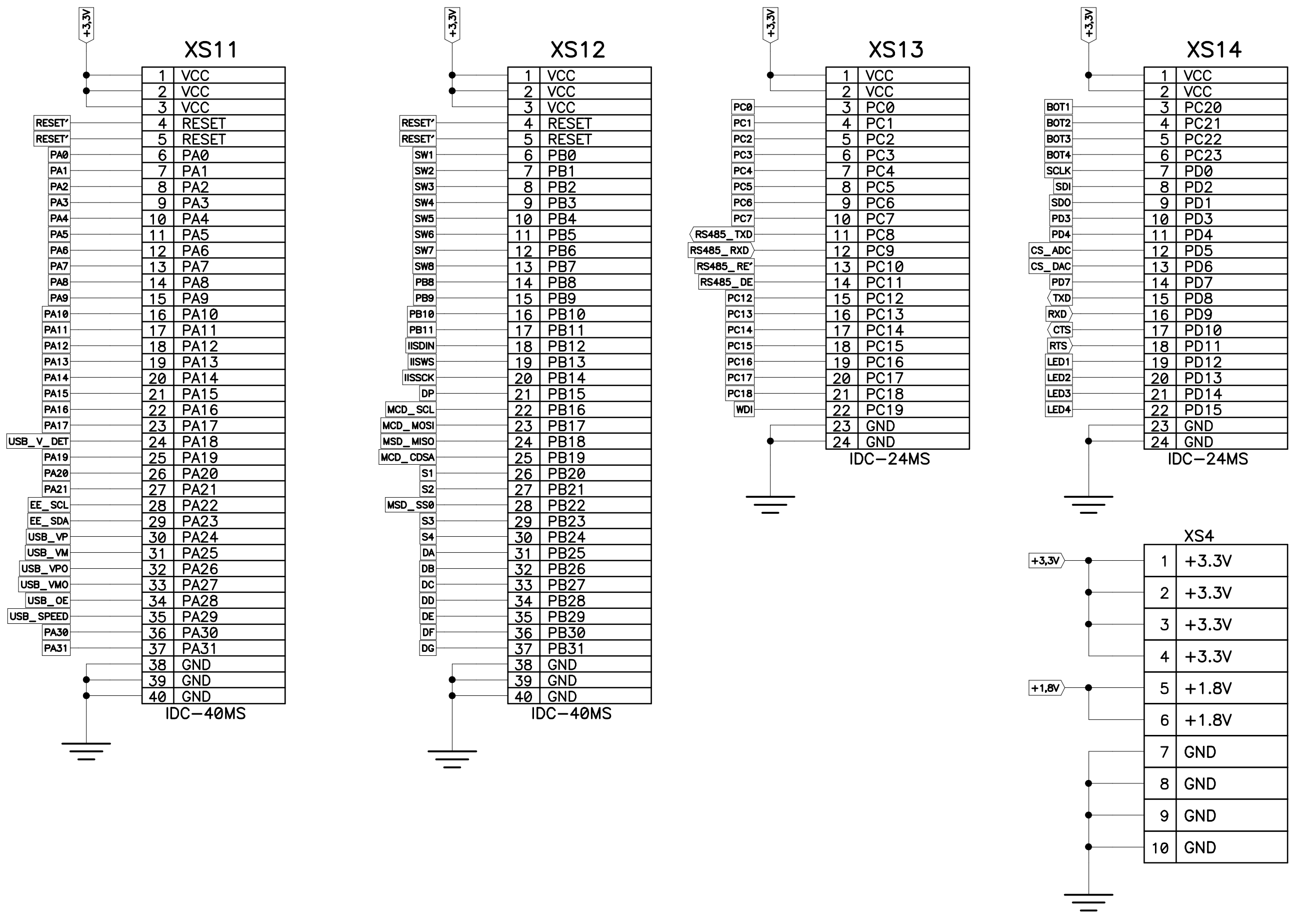
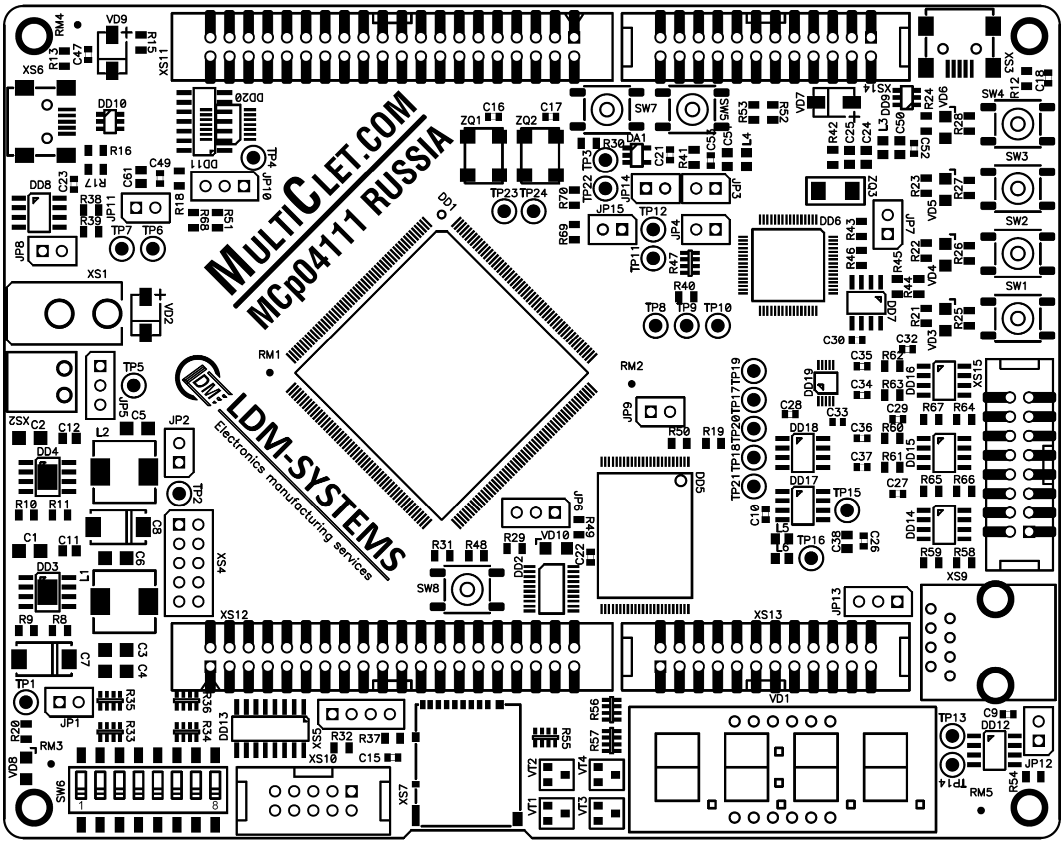


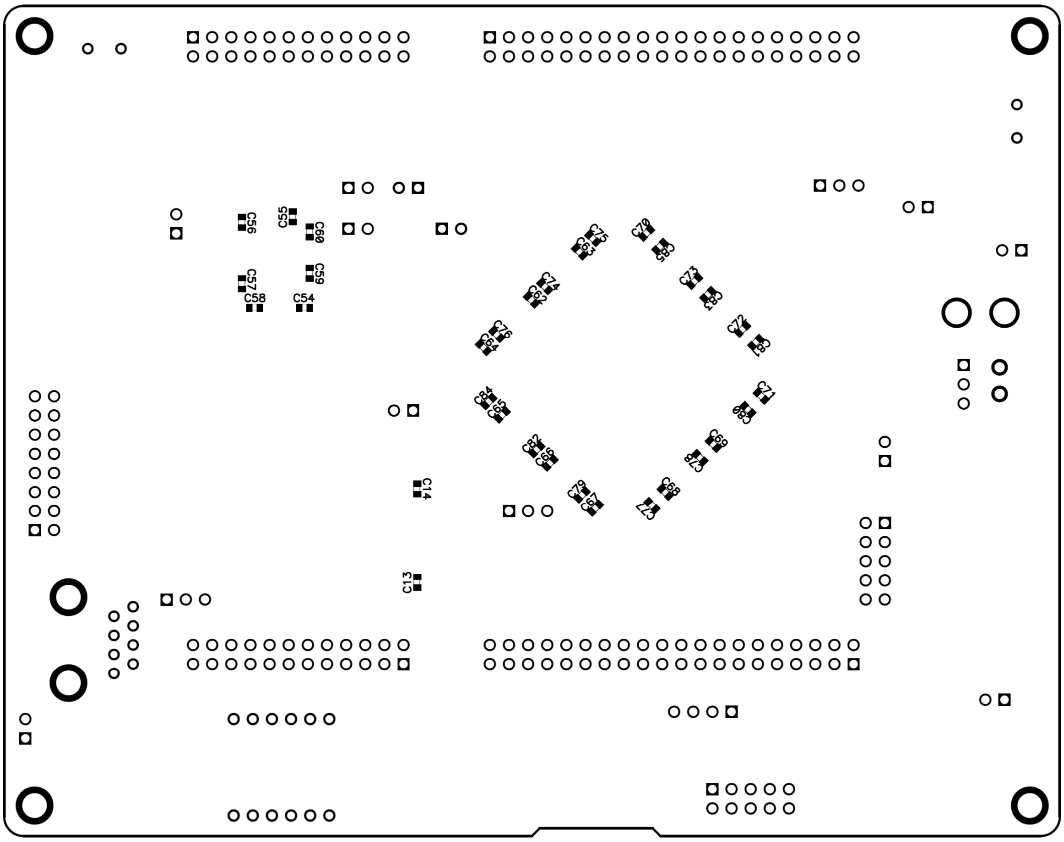
Figure 10. Electrical scheme of external connectors

2.7 Assembly scheme

TOP layer

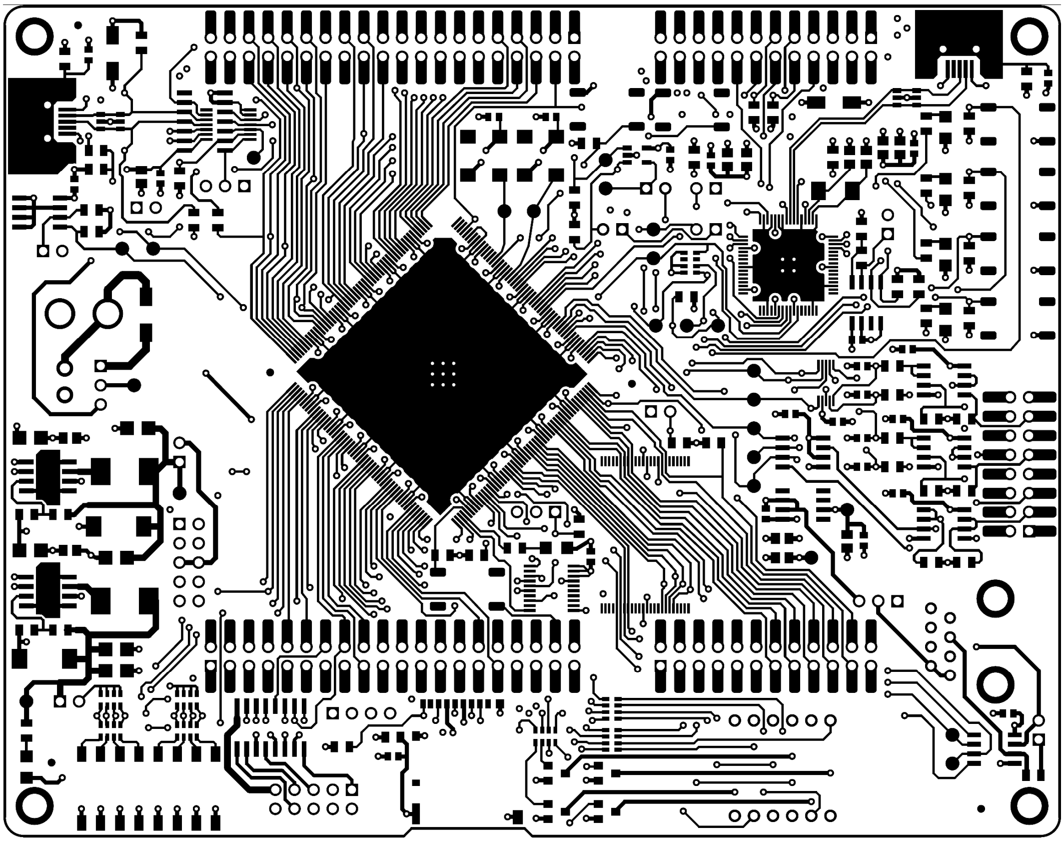


BOTTOM layer

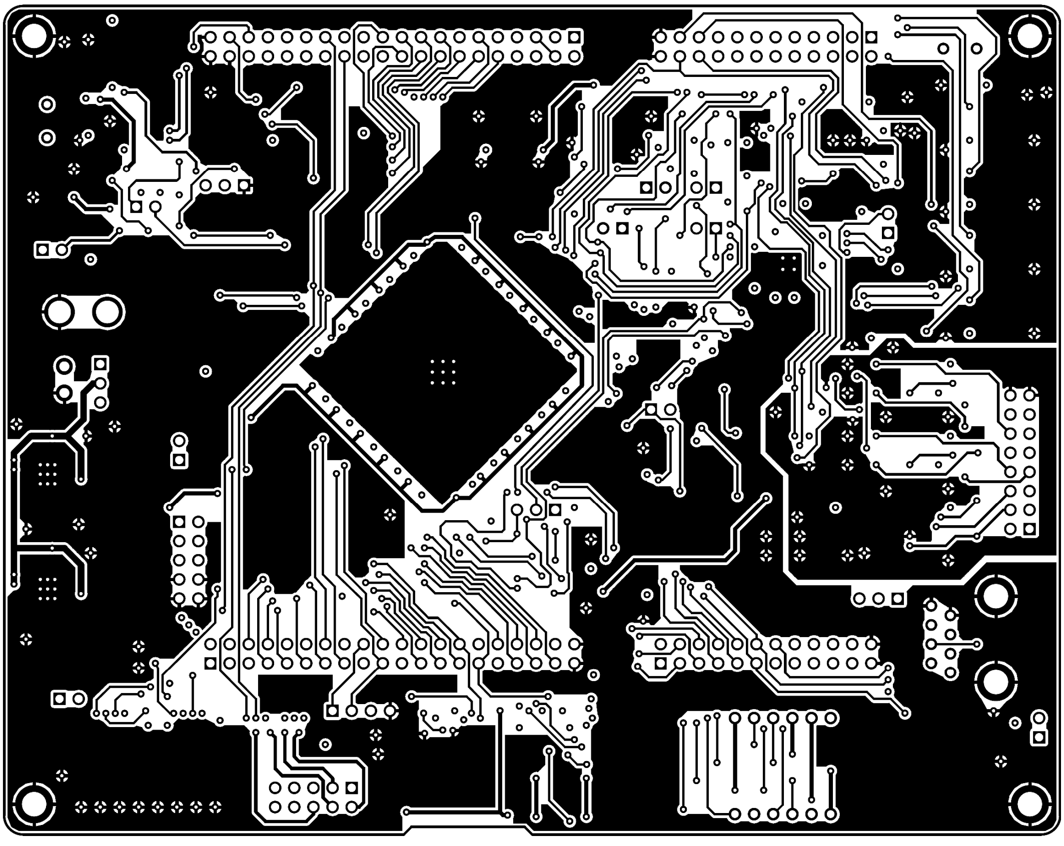


2.8 PCB layers

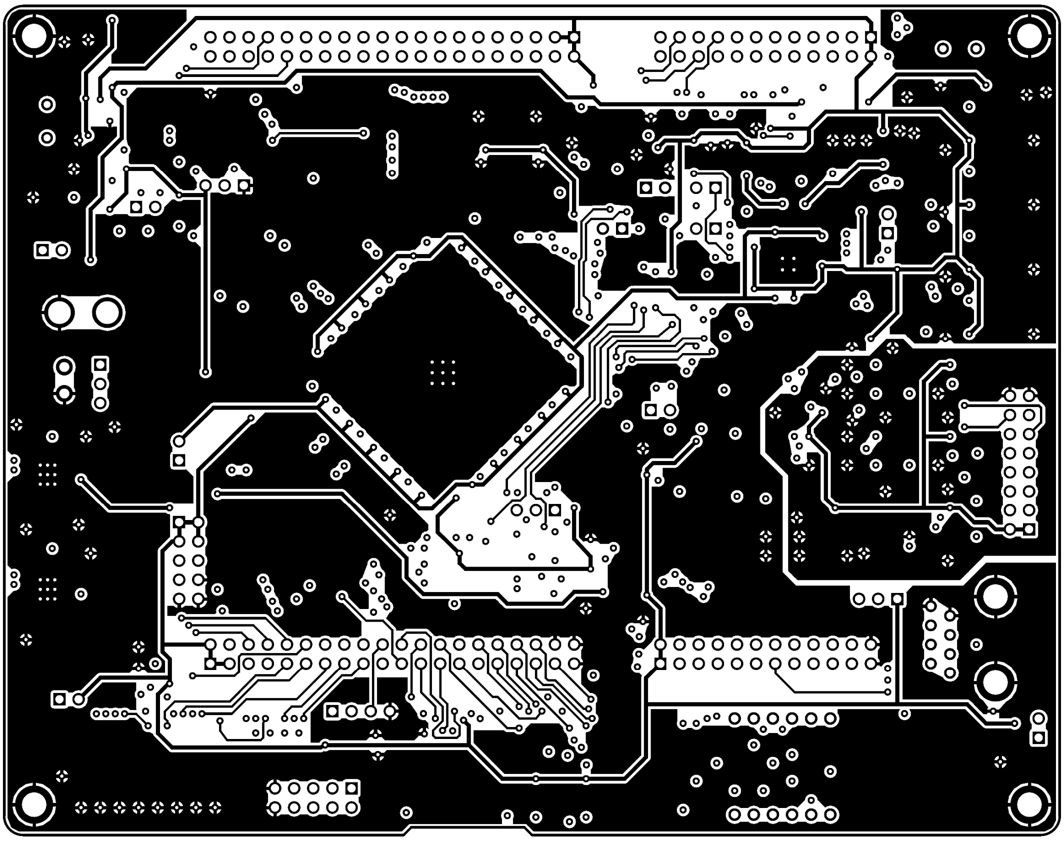
TOP layer



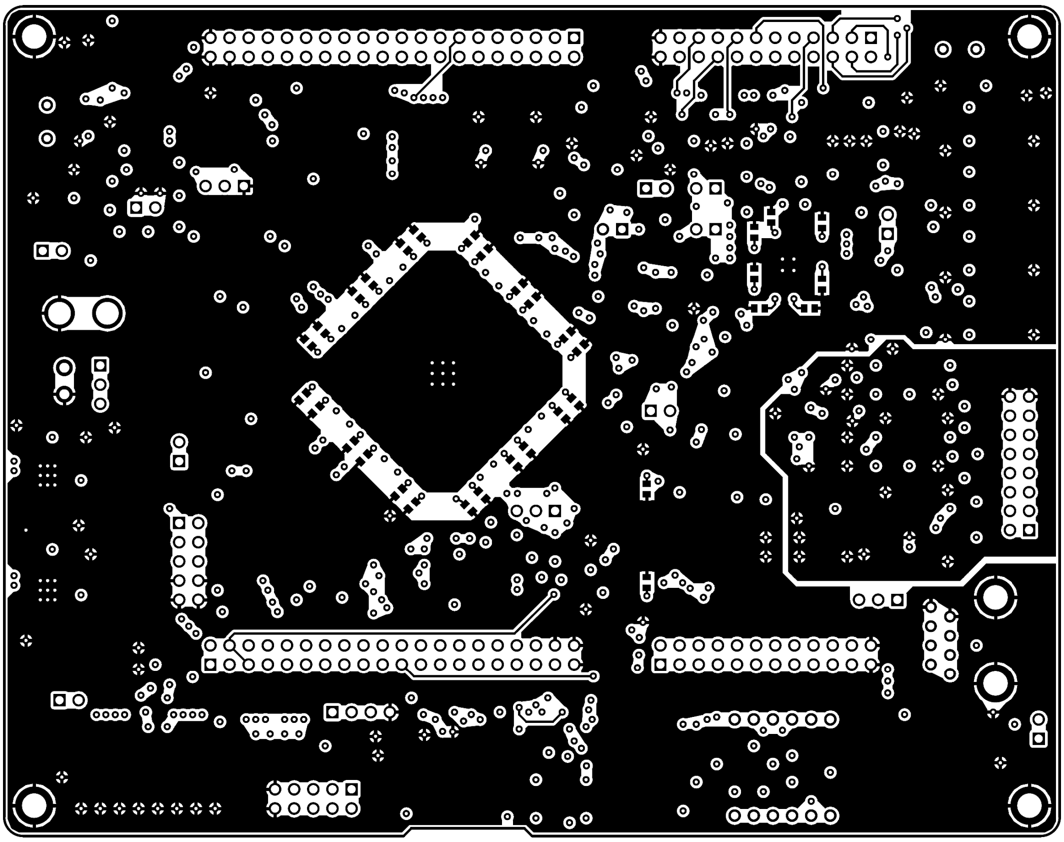
LAYER1 layer



LAYER2 layer



BOTTOM layer



3 OPERATION, KEEPING AND TRANSPORTATION

Operational environment:

Testing, transportation, storage and operation of this device do not harm the environment and human health. It keeps its parameters in the whole temperature range from 0 °C to +70 ˚C with a relative humidity less than 80%, without-condensing, when the primary power supply voltage within acceptable limits. EMC product complies with all requirements for this equipment class. Produced asymmetrical interference voltage does not exceed the allowable values in accordance with government standart 51318.22-99.

Storage conditions:

The product should be stored in warehouses, protected from the effects of precipitation, on the shelves in the same order in the absence of vapors of acids, alkalis and other corrosive materials. Conditions of storage products in accordance with with government standart 15150-69: air temperature 5 to 40 ˚ C, relative humidity 80% at 25 ˚ C. Limit in these conditions - three years.

Transport conditions:

Transportation of the product is permitted in manufacturer's packaging - all kinds of transport, except for non-pressurized compartments of the aircraft, without distance limitations.

Transportation of packaged products can be made in covered wagons and cars, the holds of ships and aircraft cockpits sealed with an air temperature from -20 to to +70 ˚C. For any means of transportation provide the mounting fixture to the body - mount box (platform, etc).