



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ



Управление портами микроконтроллера в режиме ввода-вывода общего назначения (GPIO)

**Преподаватель:
Олег Чеславович Ролич
К.Т.Н., доцент**



Согласно приведённым в данной презентации основным файлам, построить проекты MMVP_LW1_GPIO_OUT, MMVP_LW1_GPIO_IN и MMVP_LW1_GPIO_OUT_REGS управления портами ввода-вывода общего назначения и обеспечить возможность их пошаговой отладки **2**

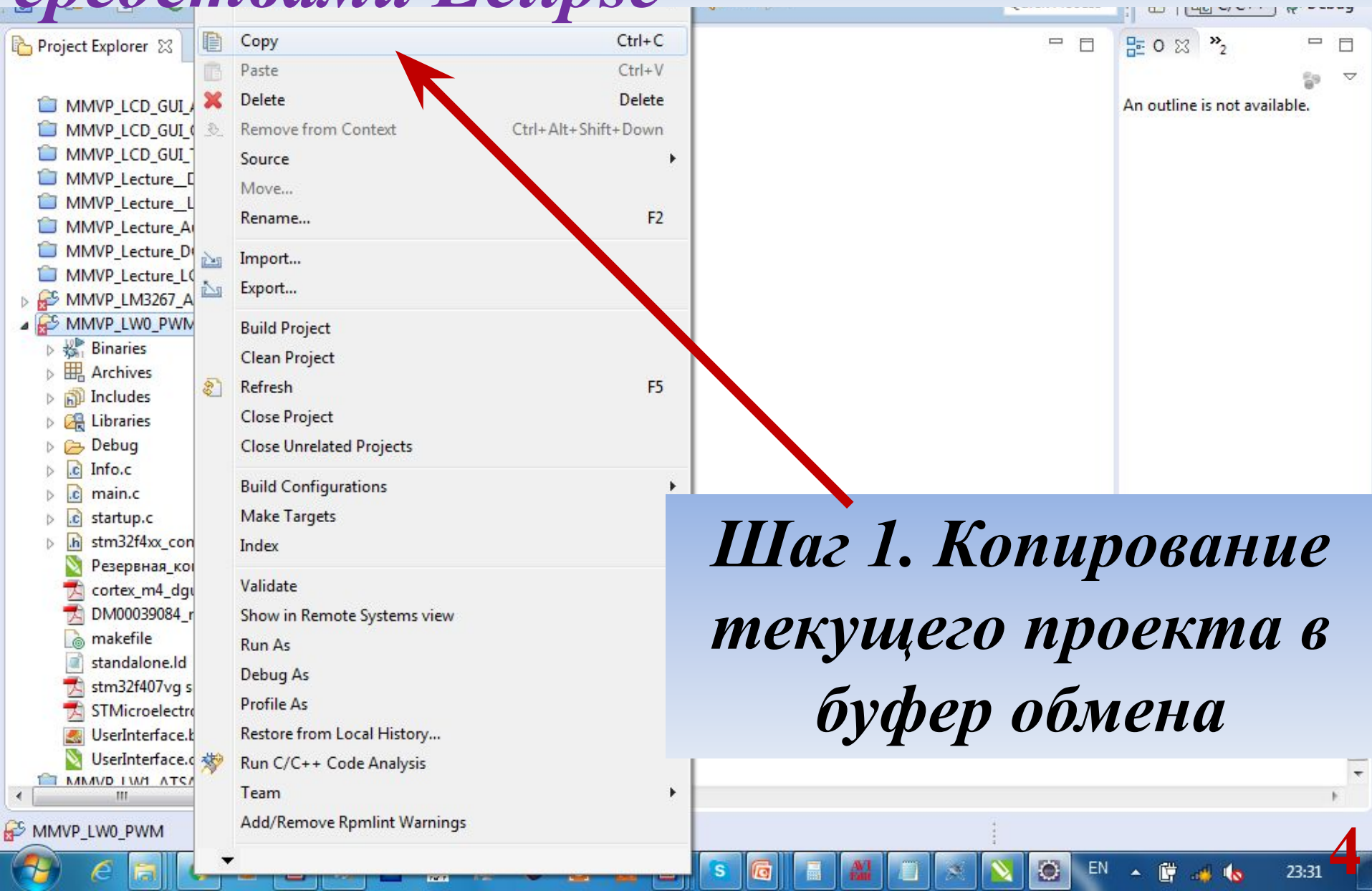


Новый проект рекомендуется создавать на базе одного из предыдущих, ранее отлаженных, например, на основе «MMVP_LW0_PWM» путём его копирования через буфер обмена (см. слайды 4 – 10)

Основы управления средствами Eclipse



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ

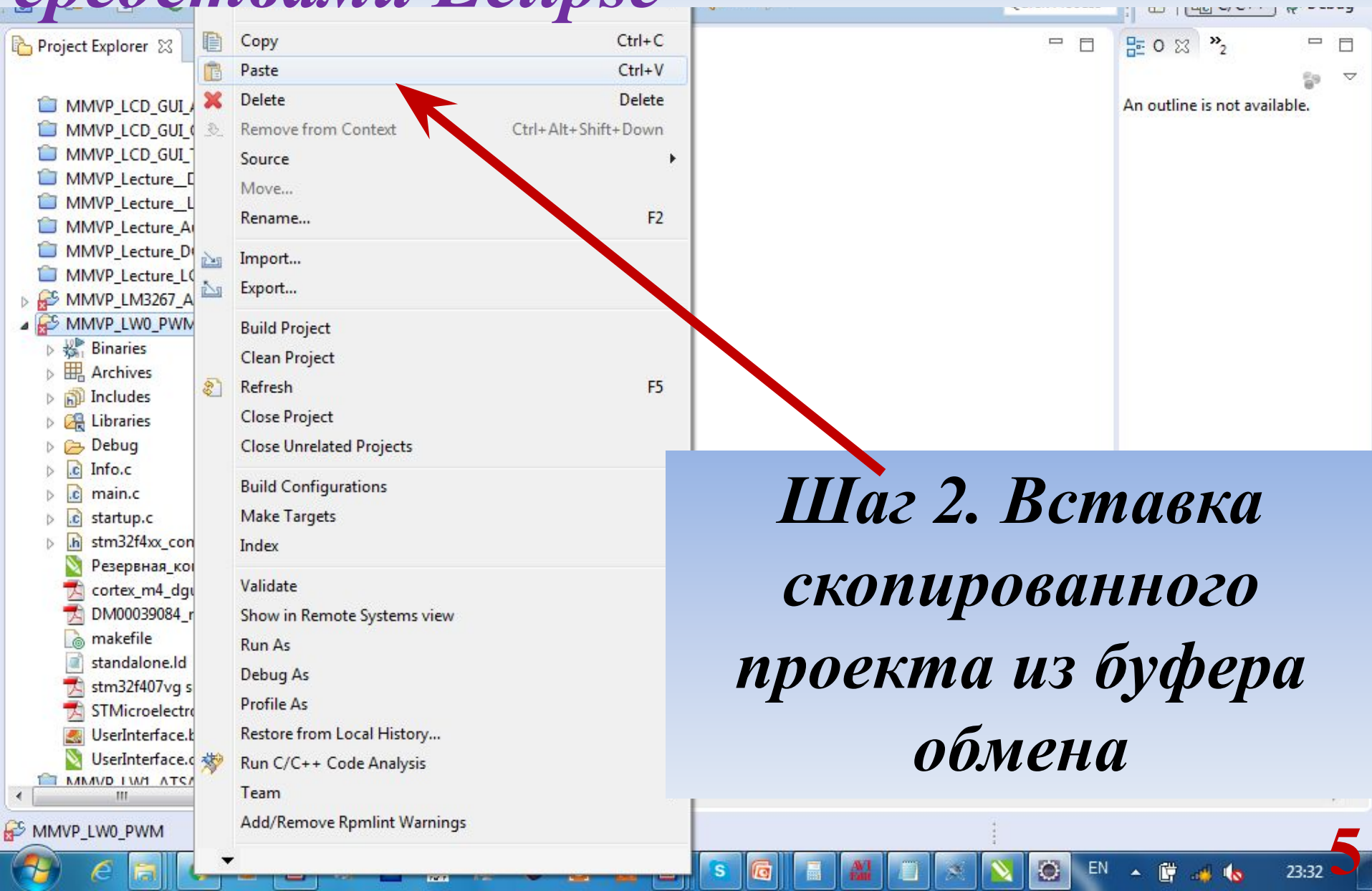


**Шаг 1. Копирование
текущего проекта в
буфер обмена**

Основы управления средствами Eclipse



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ

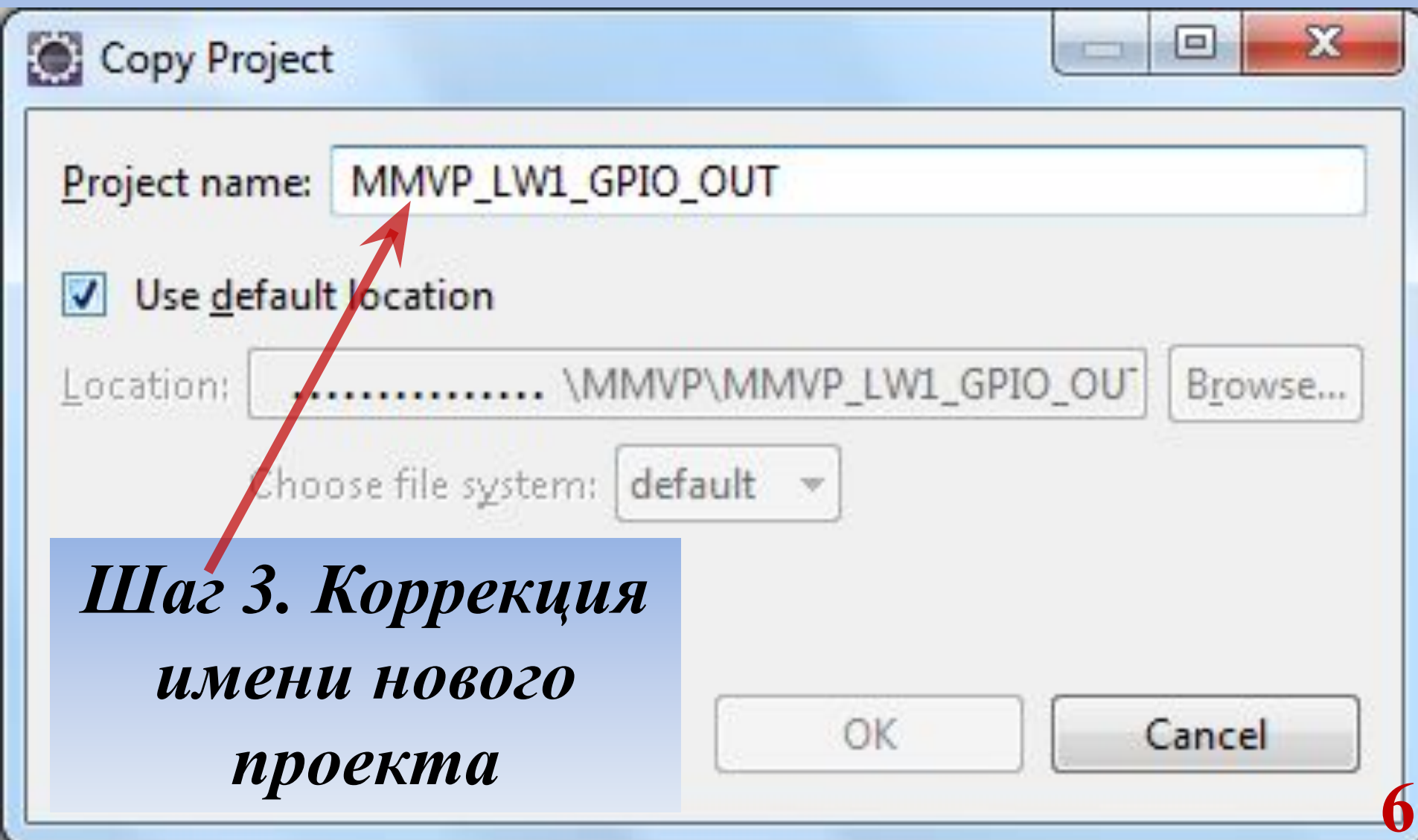


**Шаг 2. Вставка
скопированного
проекта из буфера
обмена**

Основы управления средствами Eclipse



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ



**Шаг 3. Коррекция
имени нового
проекта**

Основы управления средствами Eclipse



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ



Project Explorer

- MMVP_Lecture_LCD_To
- MMVP_LM3267_ATMEC
- MMVP_LW0_PWM
- MMVP_LW1_ATSAM3N
- MMVP_LW1_GPIO_IN
- MMVP_LW1_GPIO_OUT**
 - Binaries
 - Archives
 - Includes
 - Libraries
 - Debug
 - Info.c
 - main.c
 - startup.c
 - stm32f4xx_conf.h
 - Резервная копия_U
 - cortex_m4_dgug.pdf
 - DM00039084_new.p
 - makefile
 - standalone.ld
 - stm32f407vg small c
 - STMicroelectronics-
 - UserInterface.bmp
 - UserInterface.cdr
 - MMVP_LW1_GPIO_OUT
 - MMVP_LW11_ATSAM3I
 - MMVP_LW12_ATMEGA
 - MMVP_MEMS_MIC_SPI

Context Menu:

- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete (Delete)
- Remove from Context (Ctrl+Alt+Shift+Down)
- Source
- Move...
- Rename... (F2)
- Import...
- Export...
- Build Project
- Clean Project** (F5)
- Refresh
- Close Project
- Close Unrelated Projects
- Build Configurations
- Make Targets
- Index
- Validate
- Show in Remote Systems view
- Run As
- Debug As
- Profile As
- Restore from Local History...
- Run C/C++ Code Analysis
- Team
- Add/Remove RpmLint Warnings

Main Editor:

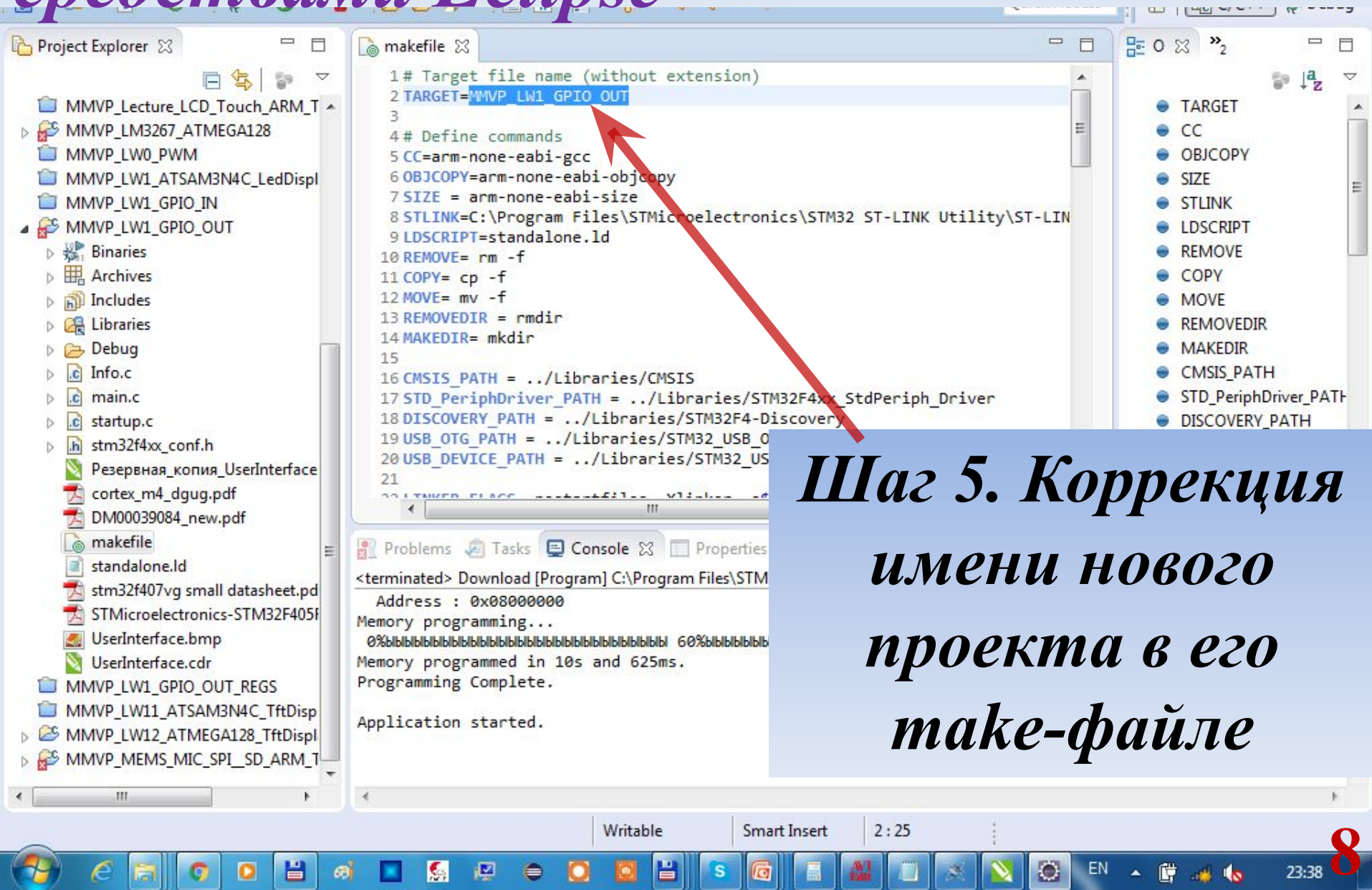
An outline is not available.

Properties: Search

STMicroelectronics\STM32 ST-LINK Utility\ST-LINK Utility\ST-LINK_CLI.exe

Шаг 4. Очистка нового проекта

MMVP_LW1_GPIO_OUT



Основы управления средствами Eclipse



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ



Project Explorer

- MMVP_Lecture_LCD_T
- MMVP_LM3267_ATME
- MMVP_LW0_PWM
- MMVP_LW1_ATSAM3M
- MMVP_LW1_GPIO_IN
- MMVP_LW1_GPIO_OUT
- Binaries
- Archives
- Includes
- Libraries
- Debug
- Info.c
- main.c
- startup.c
- stm32f4xx_conf.h
- Резервная копия_
- cortex_m4_dgug.po
- DM00039084_new.p
- makefile
- standalone.ld
- stm32f407vg small
- STMicroelectronics
- UserInterface.bmp
- UserInterface.cdr
- MMVP_LW1_GPIO_OUT
- MMVP_LW11_ATSAM3
- MMVP_LW12_ATMEGA
- MMVP_MEMS_MIC_SP

Copy Ctrl+C

Paste Ctrl+V

Delete Delete

Remove from Context Ctrl+Alt+Shift+ Down

Source

Move...

Rename... F2

Import...

Export...

Build Project

Clean Project

Refresh F5

Close Project

Close Unrelated Projects

Build Configurations

Make Targets

Index

Validate

Show in Remote Systems view

Run As

Debug As

Profile As

Restore from Local History...

Run C/C++ Code Analysis

Team

Add/Remove RpmLint Warnings

electronics\STM32 ST-LINK Utility\ST-LINK

es/STM32F4xx_StdPeriph_Driver

2F4-Discovery

USB_OTG_Driver

32_USB_Device_Library

TARGET

CC

OBJCOPY

SIZE

STLINK

LDSCRIPT

REMOVE

COPY

MOVE

REMOVEDIR

MAKEDIR

CMSIS_PATH

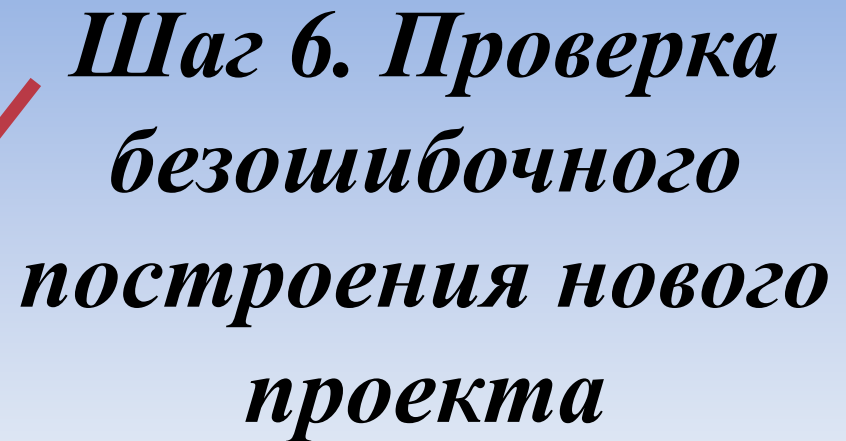
STD_PeriphDriver_PATH

DISCOVERY_PATH

USB_OTG_PATH

23:39

**Шаг 6. Проверка
безошибочного
построения нового
проекта**



Программа MMVP_LW1_GPIO_OUT *управления портами GPIO*

Файлы

info.c, startup.c, stm32f4xx_conf.h
и standalone.ld

остаются без изменений по
сравнению с проектом
«MMVP_LW0_PWM»

Программа MMVP_LW1_GPIO_OUT *управления портами GPIO*

Файл main.c

```
/*  
 * main.c  
 *  
 * Created on: 15.10.2014  
 * Author: Oleg  
 */  
  
#include "stm32f4xx_conf.h"  
  
volatile int32_t ITM_RxBuffer;  
  
RCC_ClocksTypeDef RCC_Clocks;
```


Программа MMVP_LW1_GPIO_OUT *управления портами GPIO*

```
const uint16_t arGPIO_Pins[] = {  
    GPIO_Pin_12,  
    GPIO_Pin_14,  
    GPIO_Pin_15,  
};
```

Файл main.c

```
extern const unsigned char gImage_Info[];  
unsigned short InfoImage_coor[] = {0, 0, 0, 0};
```

```
void ShowImage ( const unsigned char * pImage,  
unsigned short * pCoordinates );
```

Программа MMVP_LW1_GPIO_OUT управления портами GPIO

Файл *main.c*

```
int main (void) {  
    uint8_t uiIndex = 0;  
    GPIO_InitTypeDef      GPIO_InitStructure;  
  
    InfoImage_coor[2] = ((unsigned short)gImage_Info[3] << 8) + gImage_Info[2];  
    InfoImage_coor[3] = ((unsigned short)gImage_Info[5] << 8) + gImage_Info[4];  
  
    /* SysTick end of count event each 1 ms */  
    RCC_GetClocksFreq ( &RCC_Clocks );  
    SysTick_Config ( RCC_Clocks.HCLK_Frequency / 1000 );  
  
    STM32f4_Discovery_LCD_Init();  
    ShowImage ( gImage_Info, InfoImage_coor );  
}
```

Программа MMVP_LW1 Файл *main.c* управления портами GPIO

```
RCC_AHB1PeriphClockCmd (  
RCC_AHB1Periph_GPIOD, ENABLE );  
  
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_OUT;  
GPIO_InitStruct.GPIO_OType = GPIO_OType_PP;  
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_12 |  
GPIO_Pin_14 | GPIO_Pin_15;  
GPIO_InitStruct.GPIO_PuPd = GPIO_PuPd_UP;  
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_2MHz;  
GPIO_Init ( GPIOD, &GPIO_InitStruct );
```

Программа MMVP_LW1_GPIO_OUT *управления портами GPIO*

Файл main.c

```
while (1) {  
    Delay ( 1000 );  
    GPIO_ToggleBits ( GPIOD, arGPIO_Pins [   
uiIndex++ ] );  
    uiIndex %= ( sizeof(arGPIO_Pins) /   
sizeof(uint16_t));  
}  
return 1;  
}
```


Программа MMVP_LW1 Файл *main.c* управления портами GPIO

```
//\brief Show small image
void ShowImage ( const unsigned char * pImage, unsigned short * pCoordinates ) {
    unsigned long ulIndex_i, ulIndex_j;

    unsigned long bmpAddress = (unsigned long) pImage + 8;
    for ( ulIndex_i = 0; ulIndex_i < pCoordinates[3]; ulIndex_i++ ) {
        LCD_WriteReg ( SSD2119_X_RAM_ADDR_REG, pCoordinates[0] );
        LCD_WriteReg ( SSD2119_Y_RAM_ADDR_REG, pCoordinates[1] + ulIndex_i );
        LCD_WriteReg ( SSD2119_RAM_DATA_REG, *(__IO uint16_t *) bmpAddress );
        for ( ulIndex_j = 0; ulIndex_j < pCoordinates[2]; ulIndex_j++ ) {
            LCD_WriteRAM ( *(__IO uint16_t *) bmpAddress );
            bmpAddress += 2;
        }
    }
}
```

Программа MMVP_LW1_GPIO_OUT управления портами GPIO

Файл *makefile*

Отличительные строки make-файла

Target file name (without extension)

TARGET=MMVP_LW1_GPIO_OUT

.....

SOURCE=main.c \

startup.c \

Info.c \

\$(CMSIS_PATH)/ST/STM32F4xx/Source/Templates/system_stm32f4xx.c \

\$(STD_PeriphDriver_PATH)/src/stm32f4xx_rcc.c \

\$(STD_PeriphDriver_PATH)/src/stm32f4xx_gpio.c \

\$(STD_PeriphDriver_PATH)/src/stm32f4xx_fsmc.c \

\$(DISCOVERY_PATH)/src/stm32f4_discovery_lcd.c \

\$(DISCOVERY_PATH)/src/fonts.c



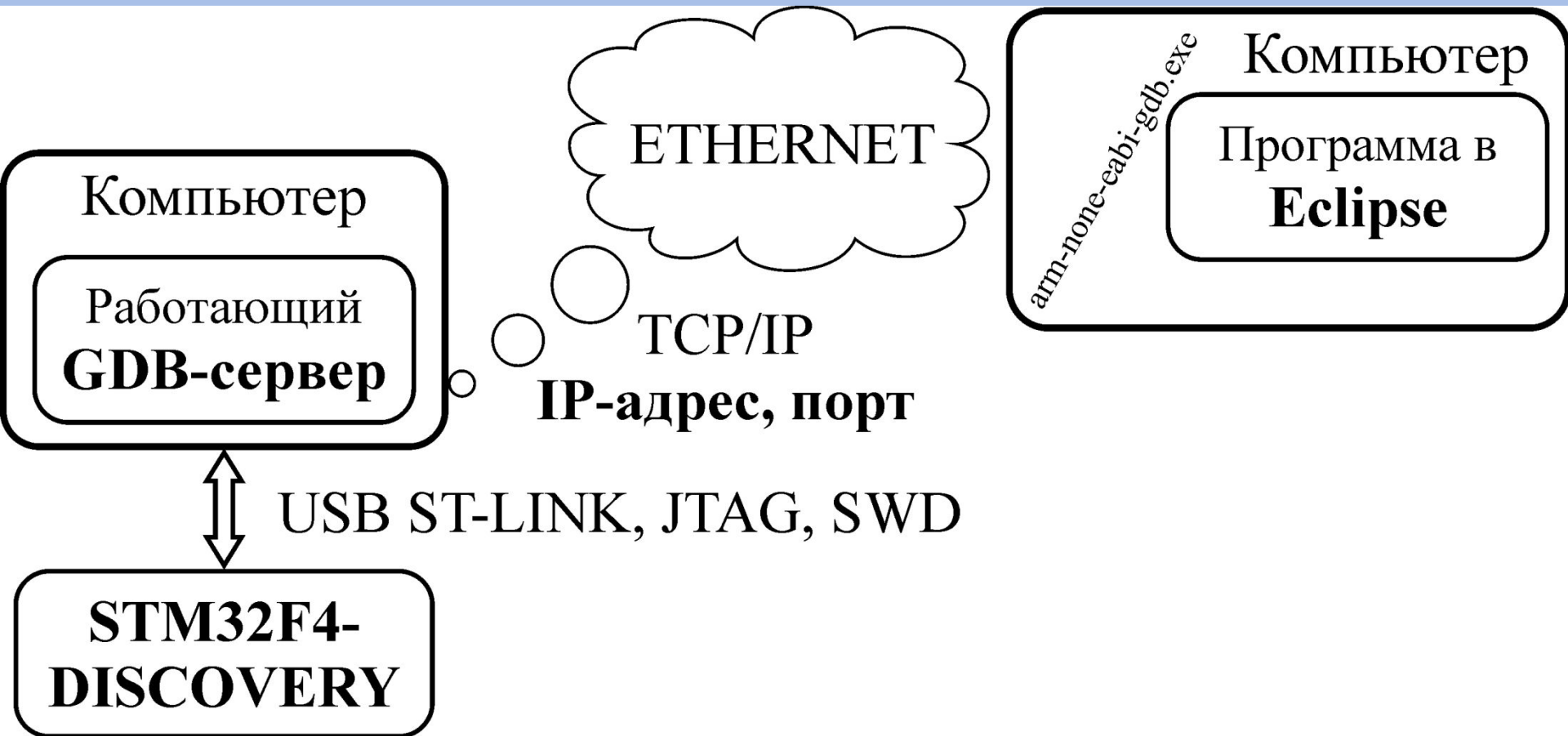
Коррекция проекта завершена

*Его необходимо
построить и загрузить
HEX-файл в контроллер,
т.е. прошить*

Технология отладки в Eclipse



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ



Технология отладки в Eclipse

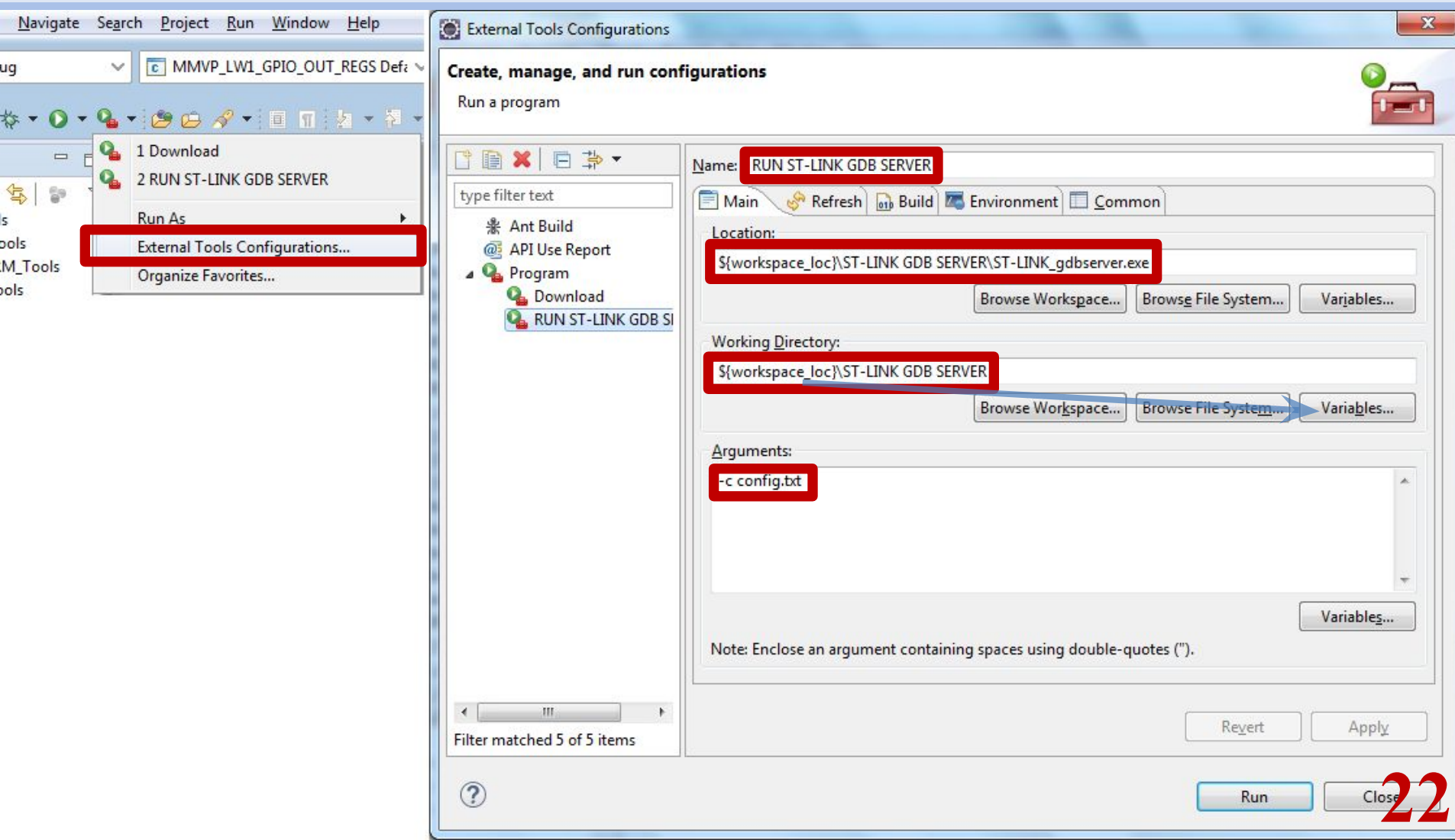


1. Настройка вызова внешней программы ST-LINK GDB SERVER
2. Настройка параметров GDB HARDWARE DEBUGGING отладки в Eclipse

Технология отладки в Eclipse

Настройка вызова внешней программы ST-LINK GDB SERVER

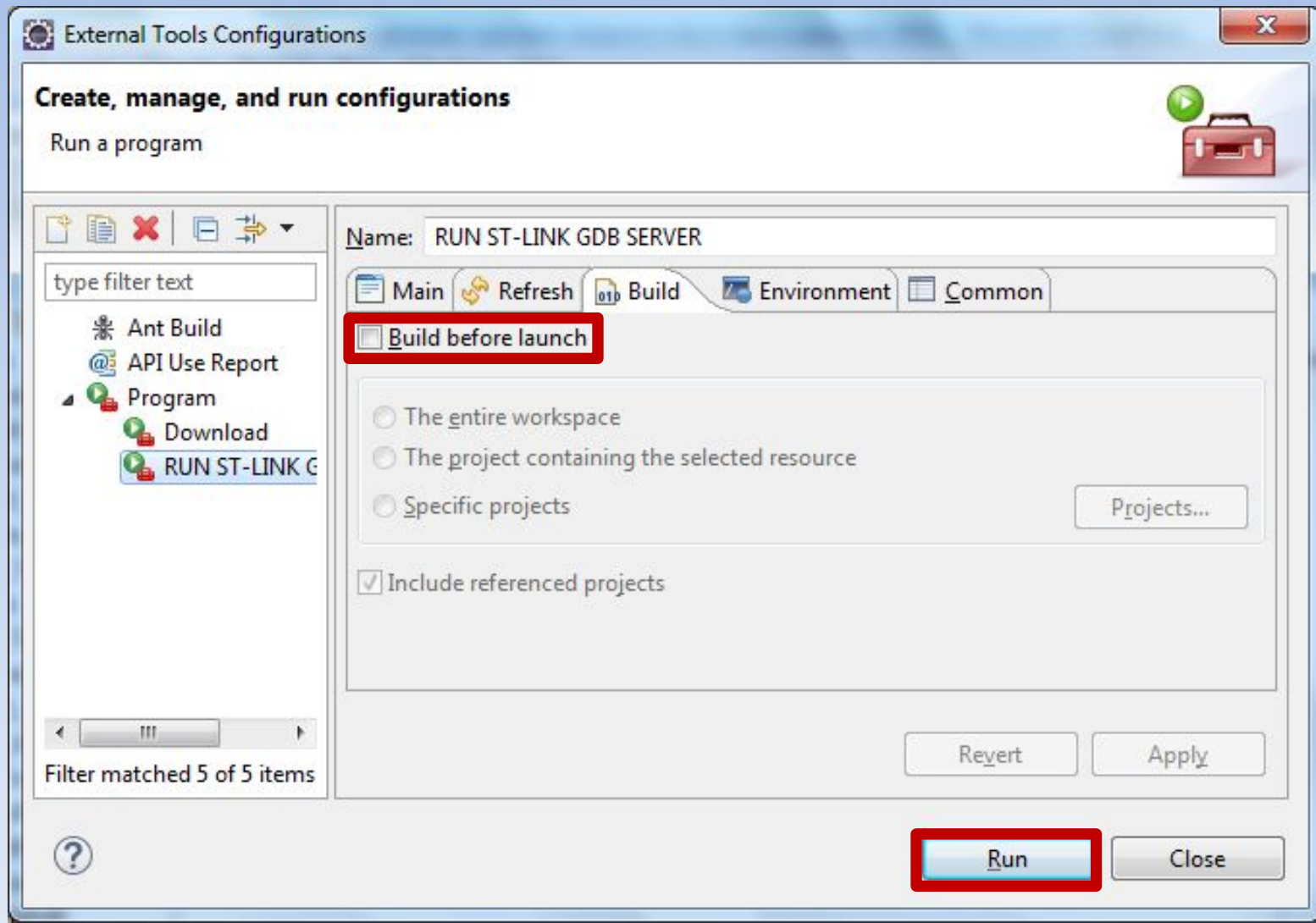
Вкладка «Main»



Технология отладки в Eclipse

Настройка вызова внешней программы ST-LINK GDB SERVER

Вкладка «Build»



Технология отладки в Eclipse



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ



Запуск GDB сервера

Вновь созданное средство
Eclipse для запуска GDB сервера

Project Explorer

- CortexM4_FreeRTOS_Acceler
- CortexM4_FreeRTOS_LCD_BM
- CortexM4_FreeRTOS_LCD_LIE
- CortexM4_FreeRTOS_LCD_Toi
- CortexM4_FreeRTOS_USB_HII
- Minimal_Project

main.c

```
MSG_END = ----- end -----
```

Makefile

1 RUN ST-LINK GDB SERVER

2 Download

Run As

External Tools Configurations...

Organize Favorites...

Problems Tasks Console Properties Search

RUN ST-LINK GDB SERVER [Program] D:\Oleg\Axonim\STLinkGDBServer\ST-LINK_gdbserver.exe

Atollic TrueSTUDIO gdbserver for ST-Link. Version 1.3.0 Pro
Developed by Atollic AB for STMicroelectronics
Copyright 2010-2011, Atollic AB and STMicroelectronics

Starting server with the following options:

Persistent Mode	: Enabled
LogFile Name	: debug_log.txt
Logging Level	: 31
Listen Port Number	: 61234
Status Refresh Delay	: 15s
Verbose Mode	: Enabled
SWD Debug	: Enabled

NVIC_DFSR_REG = 0x00000009
NVIC_CFGFSR_REG = 0x00000000
ST_LINK Major version =2 Jtag version =15
ST_LINK VID=1155 PID=14152
ST_LINK device status: HALT_MODE
Hardware watchpoint supported by the target
FuncEntry
STM32 device: id =10016413
Connected to the ST-Link Debugger.
Start Logging
Waiting for client to connect on port 61234 ...

Обратите внимание
на номер порта 61234!
Он будет фигурировать
в качестве параметра отладчика
GDB HARDWARE DEBUGGING

Результат запуска
GDB сервера

Ожидание подключения
клиента к порту 61234

Технология отладки в Eclipse

Остановка в Eclipse GDB сервера или иного процесса

Для остановки запущенных из Eclipse процессов необходимо:

1. Перейти в раздел «Debug»

2. Нажать на правую клавишу мыши в области названия процесса и выбрать в появившемся меню пункт «Terminate»

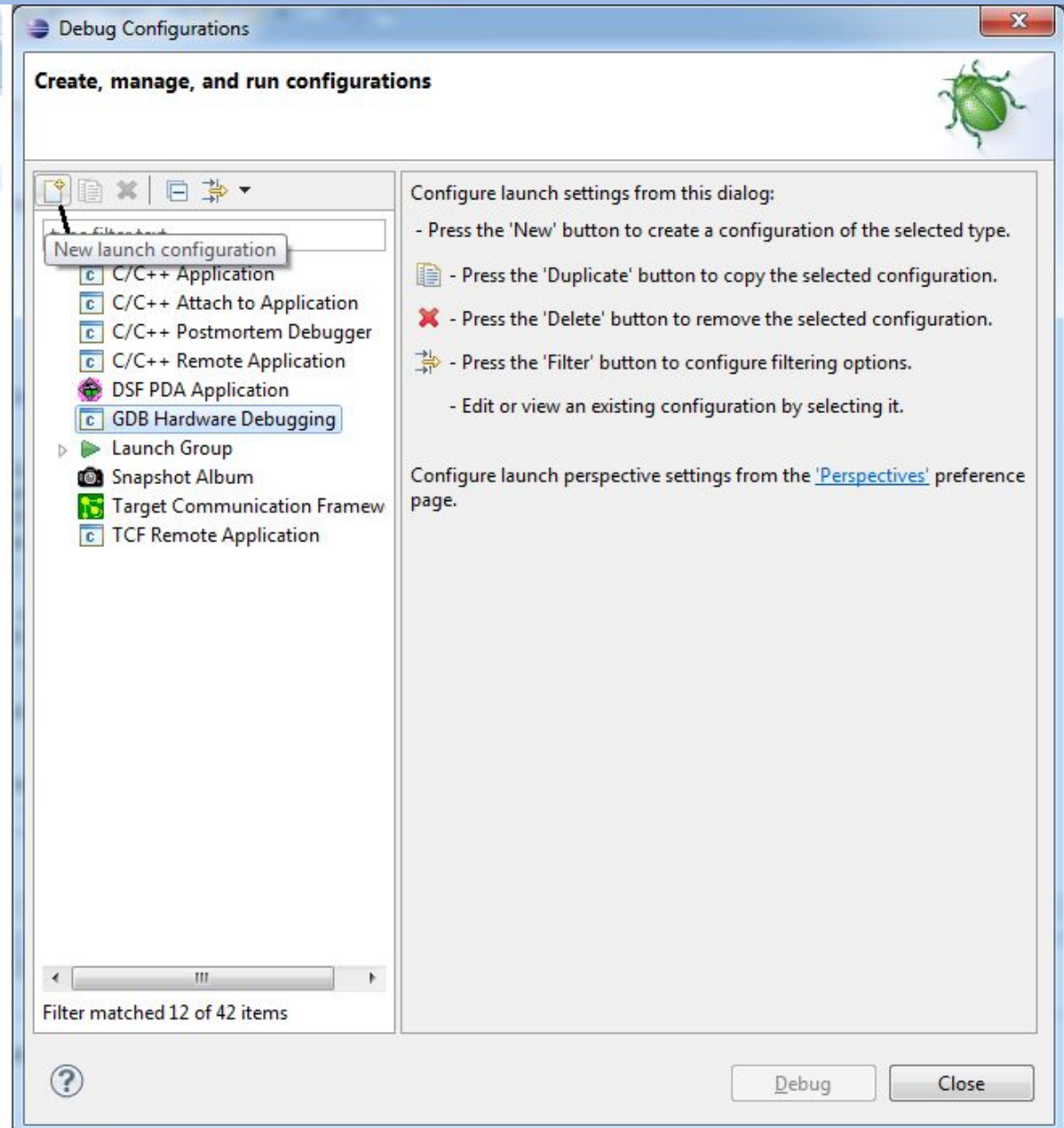
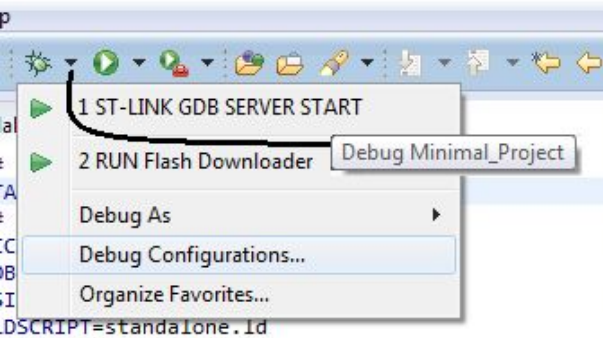
The screenshot shows the Eclipse IDE interface during a debug session. The title bar reads "Debug - Minimal_Project/Makefile - Eclipse". The menu bar includes File, Edit, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and debugging. The "Debug" tab is active, showing a list of debug elements. The first element is "RUN ST-LINK GDB SERVER [Program]" with a path "D:\Oleg\Axoni...". A right-click context menu is open over this element, with the "Terminate" option highlighted. Other options in the menu include "Copy Stack", "Find...", "Drop To Frame", "Step Into", "Step Over", "Step Return", "Instruction Stepping Mode", "Use Step Filters", "Connect...", "Resume Without Signal", "Resume", "Suspend", "Terminate and Relaunch", "Disconnect", "Remove All Terminated", "Relaunch", "Edit RUN ST-LINK GDB SERVER...", "Terminate and Remove", and "Terminate/Disconnect All". The "Properties" tab is also visible at the bottom of the menu. On the right side, the "Variables" panel shows a table with columns "Name" and "Value", containing entries for "pdev", "src", and "fifo". Below it, the "Registers" and "Memory" panels are visible. The "Console" panel at the bottom shows the output of the "Atollic TrueSTUDIO gdbserver for ST-Link. Version 1.3.0 Pro" process.

Name	Value
"pdev"	
"src"	
"fifo"	
+ Add new expression	

```
Atollic TrueSTUDIO gdbserver for ST-Link. Version 1.3.0 Pro
Developed by Atollic AB for STMicroelectronics
```

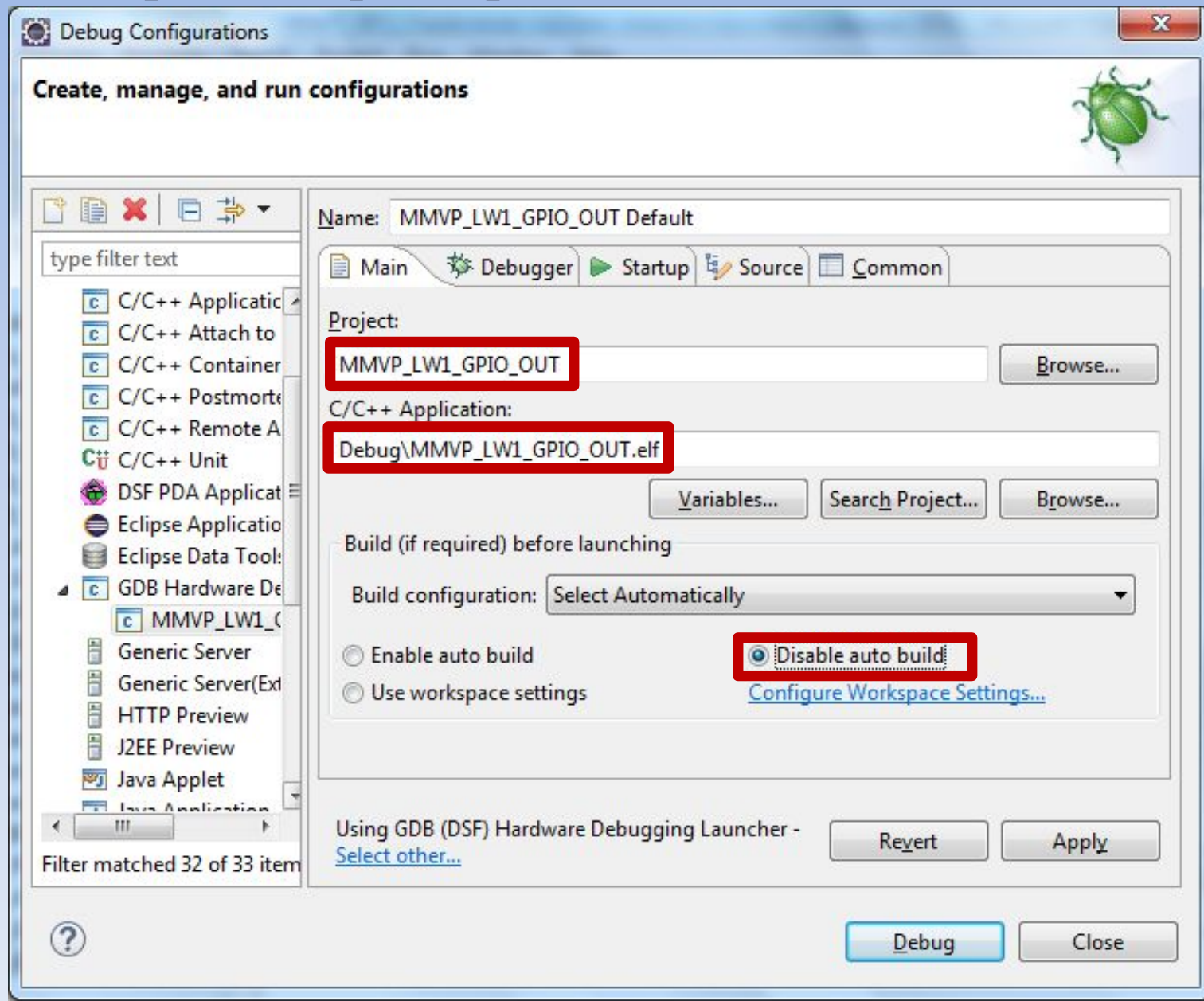

Технология отладки в Eclipse

Подготовка к настройке параметров отладки



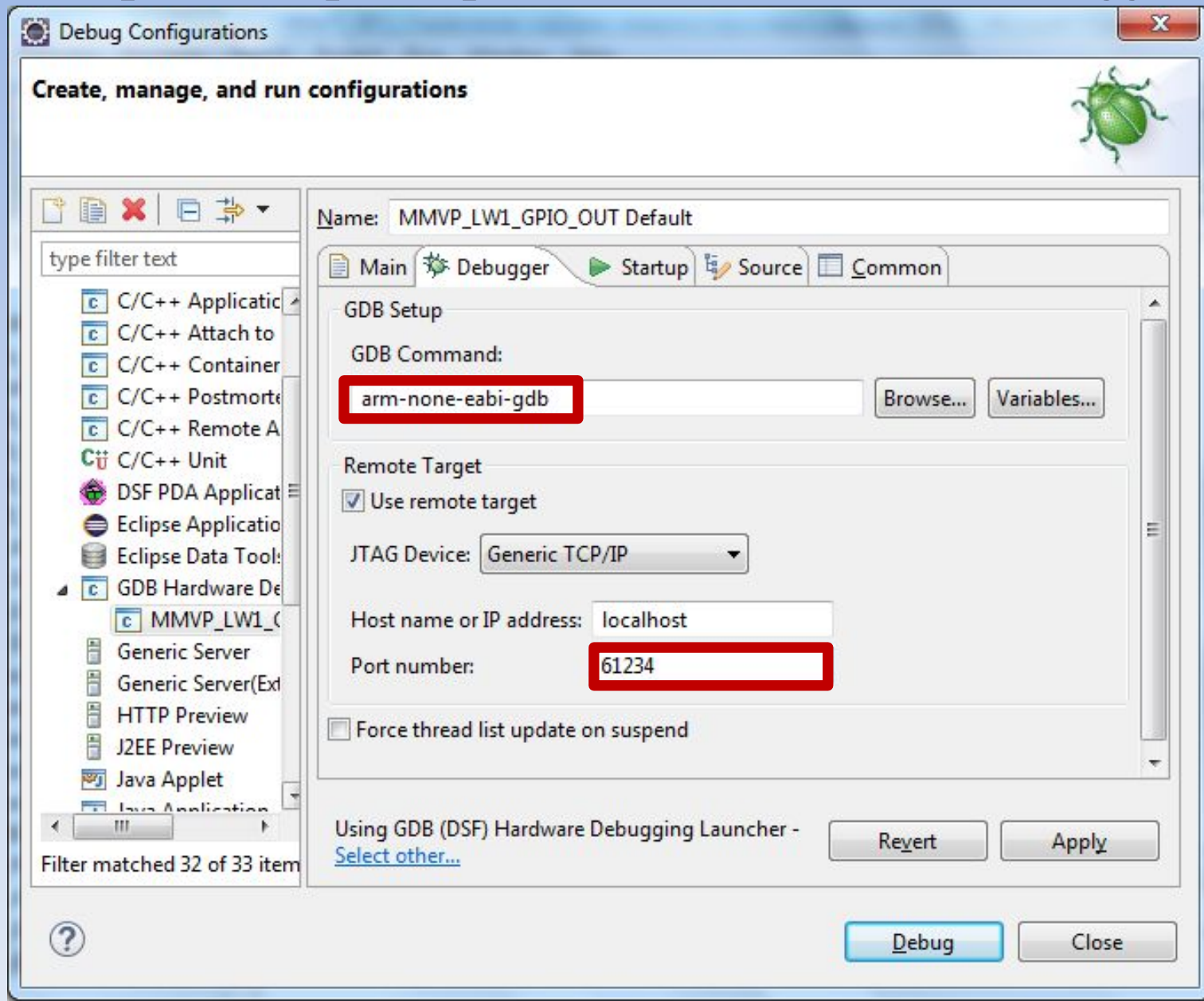
Технология отладки в Eclipse

Настройка параметров отладки. Вкладка «Main»



Технология отладки в Eclipse

Настройка параметров отладки. Вкладка «Debugger»



Create, manage, and run configurations



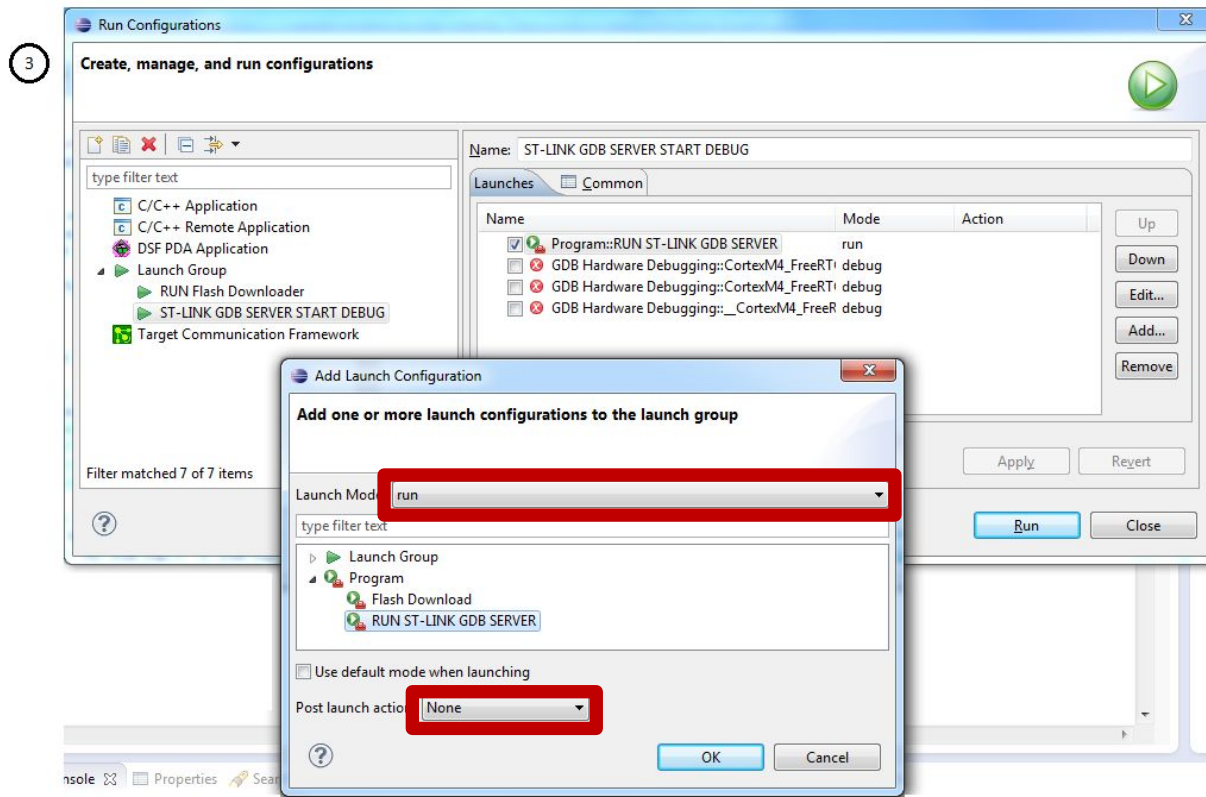
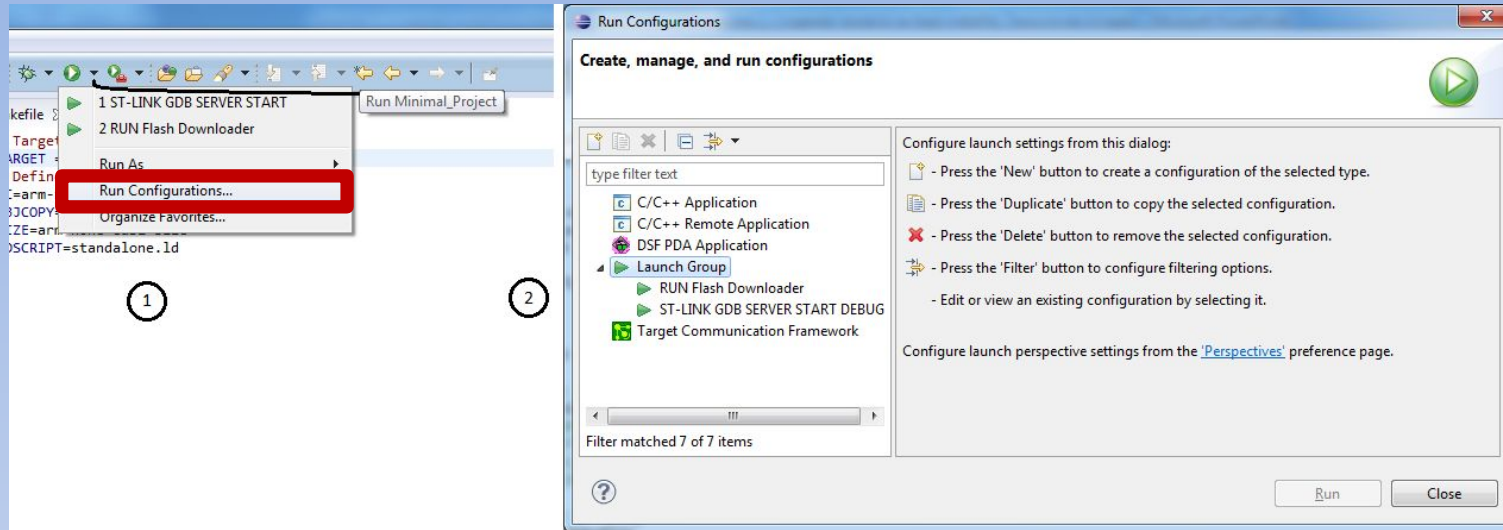
Настройка параметров отладки. Завершающая вкладка «Startup»

The screenshot shows the 'Debug Configurations' dialog in Eclipse, specifically the 'Startup' tab for a configuration named 'MMVP_LW1_GPIO_OUT Default'. The left sidebar lists various debug configurations, with 'MMVP_LW1_GPIO_OUT' selected. The main area contains several sections:

- Load Image and Symbols:**
 - ☒ Load image
 - ☒ Use project binary: MMVP_LW1_GPIO_OUT.elf
 - ☐ Use file: [text field] [Workspace...] [File System...]
 - Image offset (hex): [text field]
- Load symbols:**
 - ☒ Load symbols
 - ☒ Use project binary: MMVP_LW1_GPIO_OUT.elf
 - ☐ Use file: [text field] [Workspace...] [File System...]
 - Symbols offset (hex): [text field]
- Runtime Options:**
 - ☐ Set program counter at (hex): [text field]
 - ☒ Set breakpoint at: main

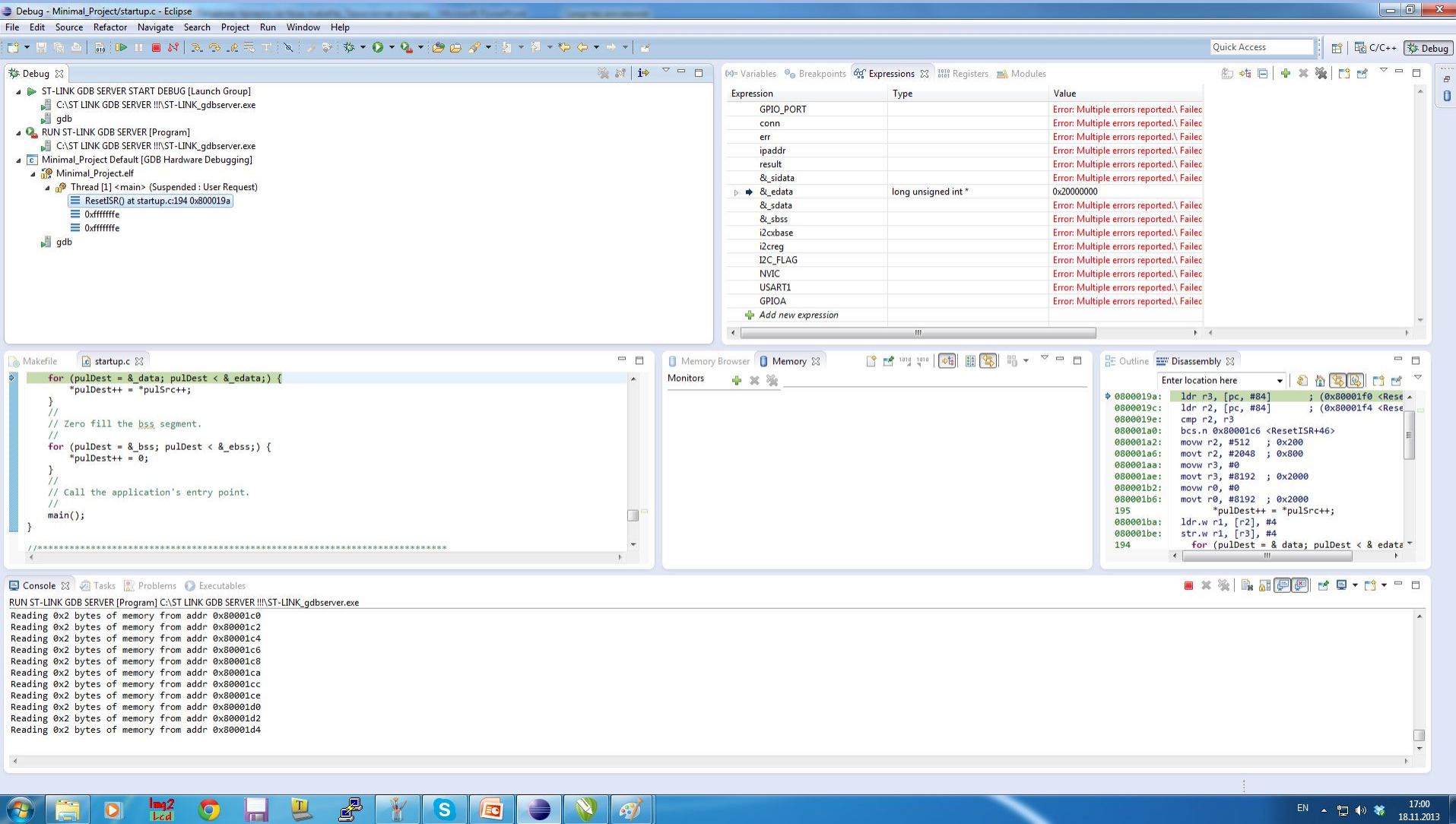
At the bottom, it says 'Using GDB (DSF) Hardware Debugging Launcher - [Select other...](#)'. There are 'Revert' and 'Apply' buttons, and a 'Debug' button at the very bottom.

Технология отладки в Eclipse



Подготовка и
настройка
средства
запуска очереди
программ

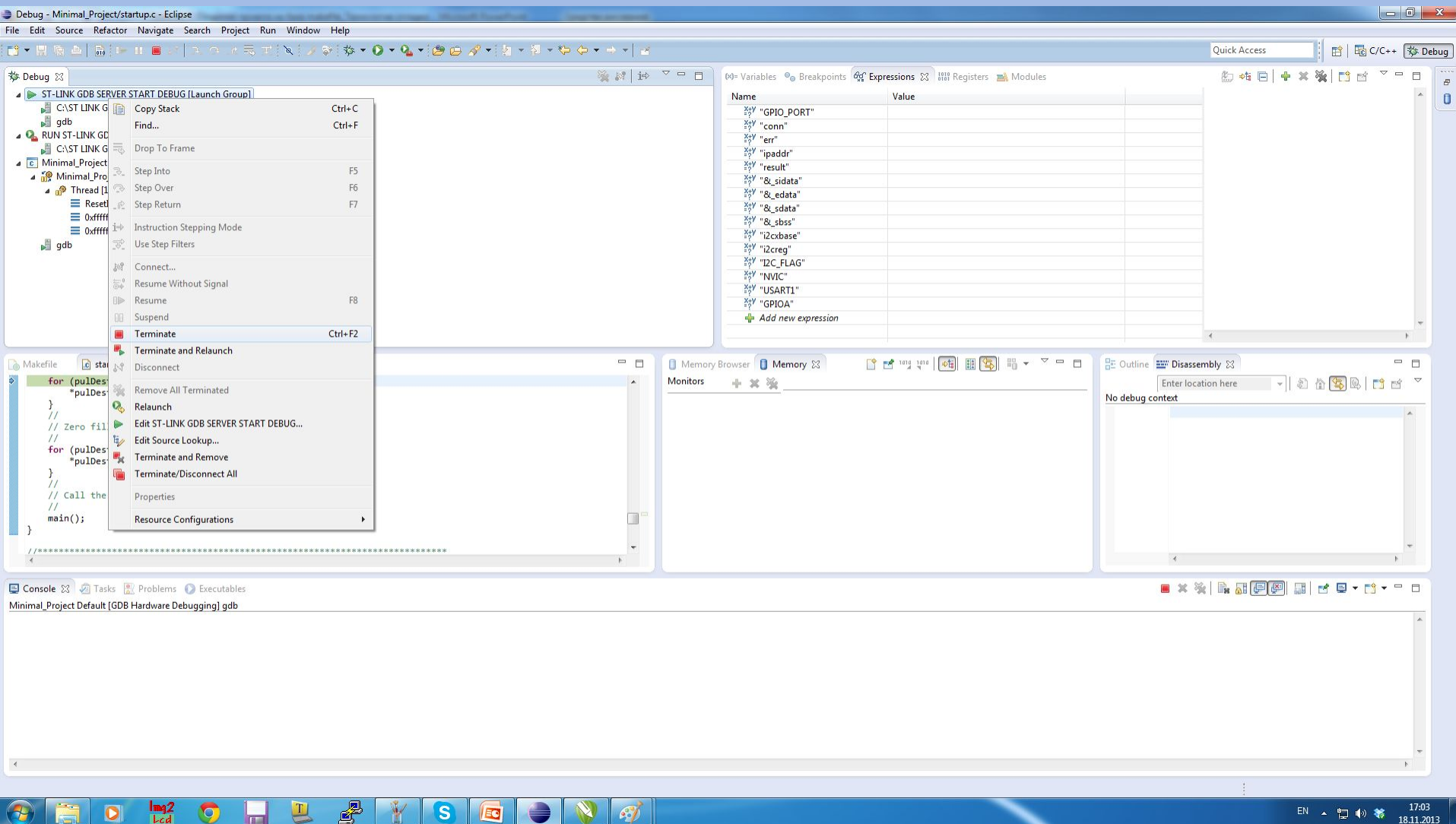
Технология отладки в Eclipse



Горячие клавиши: <F5> – шаг с входом внутрь функции, <F6> – шаг без входа в функцию, <F8> – запуск на дальнейшее выполнение до точки прерывания, <Ctrl + Shift + B> или двойной щелчок левой клавиши мыши в крайнем вертикальном поле слева – установка/удаление точки прерывания

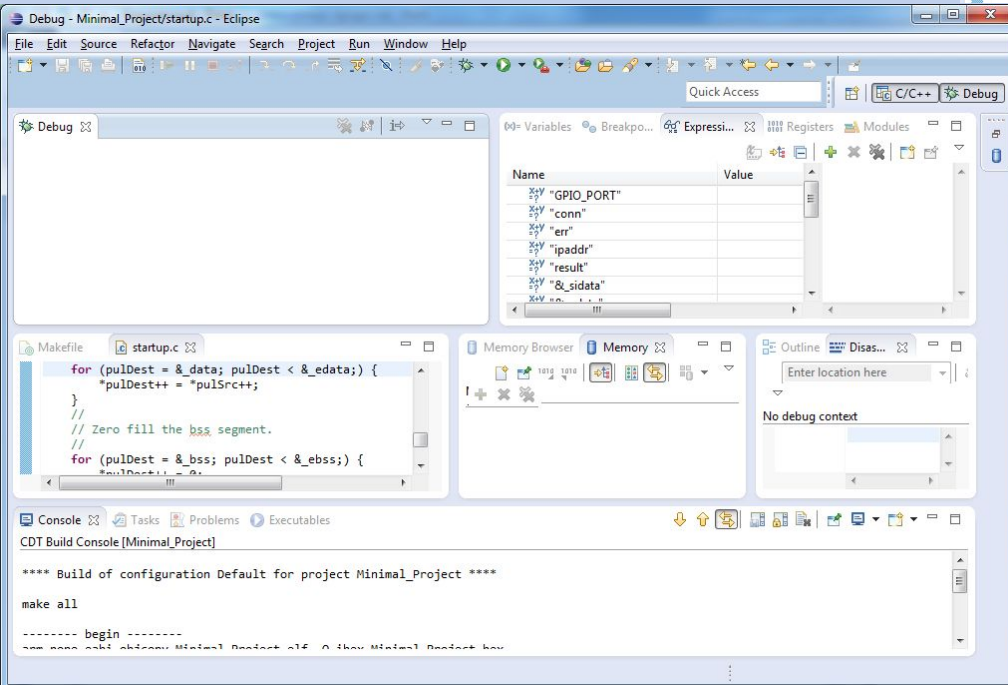
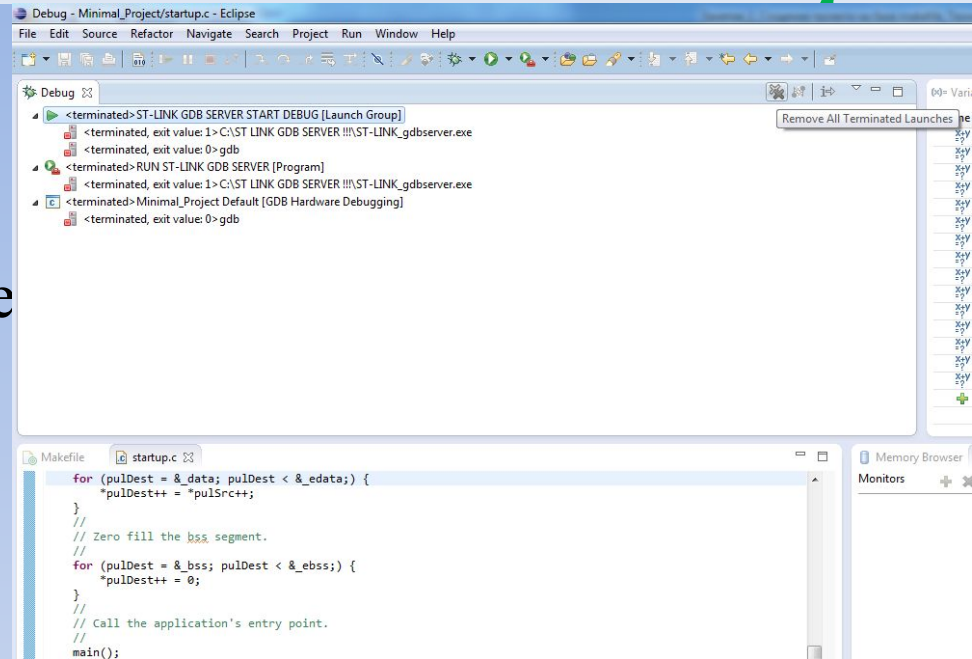
Технология отладки в Eclipse

Остановка процесса отладки (вместе с GDB сервером)



Технология отладки в Eclipse

Очистка окна «Debug» от
завершённых процессов и
переход к привычной перспективе
«C/C++»



Программа *MMVP_LW1_GPIO_IN* управления портами GPIO

Файл main.c

```
/*  
 * main.c  
 *  
 * Created on: 15.10.2014  
 * Author: Oleg  
 */  
  
#include "stm32f4xx_conf.h"  
  
volatile int32_t ITM_RxBuffer;  
  
RCC_ClocksTypeDef RCC_Clocks;
```

Программа *MMVP_LW1_GPIO_IN* управления портами GPIO

```
const uint16_t arGPIO_Pins[] = {  
    GPIO_Pin_12,  
    GPIO_Pin_14,  
    GPIO_Pin_15,  
};
```

Файл main.c

```
extern const unsigned char gImage_Info[];  
unsigned short InfoImage_coor[] = {0, 0, 0, 0};
```

```
void ShowImage ( const unsigned char * pImage,  
unsigned short * pCoordinates );
```

Программа *MMVP_LW1_GPIO_IN* управления портами GPIO

Файл *main.c*

```
int main (void) {
    uint8_t uiIndex = 0;
    GPIO_InitTypeDef      GPIO_InitStructure;

    InfoImage_coor[2] = ((unsigned short)gImage_Info[3] << 8) + gImage_Info[2];
    InfoImage_coor[3] = ((unsigned short)gImage_Info[5] << 8) + gImage_Info[4];

    /* SysTick end of count event each 1 ms */
    RCC_GetClocksFreq ( &RCC_Clocks );
    SysTick_Config ( RCC_Clocks.HCLK_Frequency / 1000 );

    STM32f4_Discovery_LCD_Init();
    ShowImage ( gImage_Info, InfoImage_coor );
```


Программа *MMVP_LW* Файл *main.c* управления портами *GPIO*

```
RCC_AHB1PeriphClockCmd (  
RCC_AHB1Periph_GPIO, ENABLE );
```

```
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_OUT;  
GPIO_InitStruct.GPIO_OType = GPIO_OType_PP;  
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_12 |  
GPIO_Pin_14 | GPIO_Pin_15;  
GPIO_InitStruct.GPIO_PuPd = GPIO_PuPd_UP;  
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_2MHz;  
GPIO_Init ( GPIO, &GPIO_InitStruct );
```

Программа *MMVP_LW* Файл *main.c* управления портами *GPIO*

```
RCC_AHB1PeriphClockCmd (  
RCC_AHB1Periph_GPIOA, ENABLE );
```

```
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_IN;  
GPIO_InitStruct.GPIO_OType = GPIO_OType_OD;
```

```
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_0;
```

```
GPIO_InitStruct.GPIO_PuPd =
```

```
GPIO_PuPd_NOPULL;
```

```
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_2MHz;
```

```
GPIO_Init ( GPIOA, &GPIO_InitStruct );
```

Программа *MMVP_LW* Файл *main.c* управления портами *GPIO*

```
while (1) {  
    if (GPIO_ReadInputDataBit ( GPIOA, GPIO_Pin_0  
) == Bit_SET) {  
        GPIO_ToggleBits ( GPIOD, arGPIO_Pins [   
uiIndex++ ] );  
        uiIndex %= (sizeof(arGPIO_Pins) /  
sizeof(uint16_t));  
        Delay ( 300 );  
    }  
}  
return 1;  
}
```

Программа *MMVP_LI* Файл *main.c* управления портами *GPIO*

```
//\brief Show small image
void ShowImage ( const unsigned char * pImage, unsigned short * pCoordinates ) {
    unsigned long ulIndex_i, ulIndex_j;

    unsigned long bmpAddress = (unsigned long) pImage + 8;
    for ( ulIndex_i = 0; ulIndex_i < pCoordinates[3]; ulIndex_i++ ) {
        LCD_WriteReg ( SSD2119_X_RAM_ADDR_REG, pCoordinates[0] );
        LCD_WriteReg ( SSD2119_Y_RAM_ADDR_REG, pCoordinates[1] + ulIndex_i );
        LCD_WriteReg ( SSD2119_RAM_DATA_REG, *(__IO uint16_t *) bmpAddress );
        for ( ulIndex_j = 0; ulIndex_j < pCoordinates[2]; ulIndex_j++ ) {
            LCD_WriteRAM ( *(__IO uint16_t *) bmpAddress );
            bmpAddress += 2;
        }
    }
}
```

Программа MMVP_LW1_GPIO_OUT_REGS управления портами GPIO

```
#include "stm32f4xx_conf.h"
```

Файл *main.c*

```
volatile int32_t ITM_RxBuffer;
```

```
RCC_ClocksTypeDef RCC_Clocks;
```

```
extern const unsigned char gImage_Info[];
```

```
unsigned short InfoImage_coord[] = {0, 0, 0, 0};
```

```
void ShowImage ( const unsigned char * pImage,  
unsigned short * pCoordinates );
```


Программа MMVP_LW1_GPIO_OUT_REGS управления портами GPIO

Файл *main.c*

```
int main (void) {
    GPIO_InitTypeDef      GPIO_InitStruct;

    InfoImage_coor[2] = ((unsigned short)gImage_Info[3] << 8) + gImage_Info[2];
    InfoImage_coor[3] = ((unsigned short)gImage_Info[5] << 8) + gImage_Info[4];

    /* SysTick end of count event each 1 ms */
    RCC_GetClocksFreq ( &RCC_Clocks );
    SysTick_Config ( RCC_Clocks.HCLK_Frequency / 1000 );

    STM32f4_Discovery_LCD_Init();
    ShowImage ( gImage_Info, InfoImage_coor );
}
```

Программа MMVP_LW1_GP

Файл main.c

управления портами GPIO

// Далее следует программный блок непосредственного регистрового управления GPIO

```
int * pGeneralPointer;
```

```
uint8_t uiIndex = 0;
```

```
const uint8_t arGPIO_Pins[] = { 12, 15, 14 };
```

```
pGeneralPointer = (int *)0x40023830;
```

```
*pGeneralPointer = 7;
```

**В программный
код внедрена
ошибка!!!**

```
pGeneralPointer = (int *)0x40020C00;
```

```
*pGeneralPointer &= ~((0b11 << (arGPIO_Pins[0] * 2))  
| (0b11 << (arGPIO_Pins[1] * 2))  
| (0b11 << (arGPIO_Pins[2] * 2)));
```

```
*pGeneralPointer |= ((0x01 << (arGPIO_Pins[0] * 2))  
| (0x01 << (arGPIO_Pins[1] * 2))  
| (0x01 << (arGPIO_Pins[2] * 2)));
```

Программа MMVP_LW1_GPIO_OUT_REGS управления портами GPIO

Файл *main.c*

```
while (1) {  
    Delay ( 1000 );  
    pGeneralPointer = (int *)0x40020C14;  
    *pGeneralPointer ^= (1 << arGPIO_Pins [  
uiIndex++ ]);  
    uiIndex %= (sizeof(arGPIO_Pins) / sizeof(uint8_t));  
}  
return 1;  
}
```

Программа MMVP_LW1_GP

Файл *main.c*

управления портами GPIO

//\brief Show small image

```
void ShowImage ( const unsigned char * pImage, unsigned short * pCoordinates ) {
    unsigned long ulIndex_i, ulIndex_j;

    unsigned long bmpAddress = (unsigned long) pImage + 8;
    for ( ulIndex_i = 0; ulIndex_i < pCoordinates[3]; ulIndex_i++ ) {
        LCD_WriteReg ( SSD2119_X_RAM_ADDR_REG, pCoordinates[0] );
        LCD_WriteReg ( SSD2119_Y_RAM_ADDR_REG, pCoordinates[1] + ulIndex_i );
        LCD_WriteReg ( SSD2119_RAM_DATA_REG, *(__IO uint16_t *) bmpAddress );
        for ( ulIndex_j = 0; ulIndex_j < pCoordinates[2]; ulIndex_j++ ) {
            LCD_WriteRAM ( *(__IO uint16_t *) bmpAddress );
            bmpAddress += 2;
        }
    }
}
```



КАФЕДРА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННО-
КОМПЬЮТЕРНЫХ СИСТЕМ



*Лабораторная работа
завершена*