

НПФ «Мехатроника-Про»

**Описание Стартового проекта TMS320F28335 в среде
Code Composer Studio 3.3**

**Загрузка стартового проекта и библиотеки блоков с помощью
Code Composer Studio и C2Prog**

Содержание

Частота основного прерывания TINT0.....	3
Описание стартового проекта	4
Модуль конфигурации оборудования «DSP2833x_Init.c».....	5
Основной модуль «main.c».....	5
Модуль конфигурации «config.h»	6
Подключение драйвера коммуникации	6
Подключение ядра MexBIOS.....	9
Настройка памяти в MexBIOS	12
Прошивка в контроллер MexBIOS kernel в плату с процессором TMS320F28335	13
Компиляция библиотеки блоков и стартового проекта	13
Загрузка с помощью C2Prog и эмулятора XDS100v1,v2	14
Загрузка с помощью Code Composer v3.3 и эмулятора XDS100 v1,v2.....	20
Загрузка стартового проекта и библиотеки блоков с помощью	
Code Composer Studio 5.4 и эмулятора XDS100v3	24
Загрузка стартового проекта и библиотеки блоков с помощью эмулятора	
SAU510-USB JTAG Emulator	32
Работа с MexBIOS Development Studio	34

Частота основного прерывания TINT0

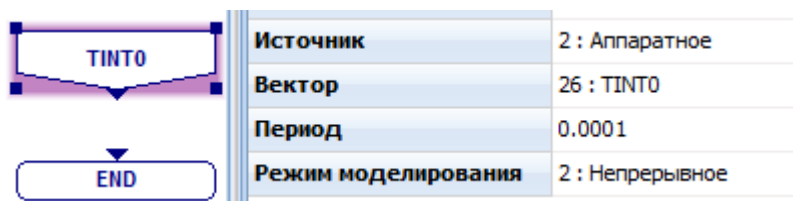
Частота основного прерывания может быть изменена в файле config.h

C:\Users\%UserName%\AppData\Roaming\NPF Mechatronica-Pro\MexBIOS Development Studio\Libraries\TMS320F2833x\StartUp\include

в строке:

```
#define HZ 10000 // Main isr frequency
```

По умолчанию период основного прерывания равен 0.0001 (частота 10 кГц) для процессоров 2833х. Именно этот период нужно задавать в настройке EVENT:



Также этот период необходимо задавать в блоки, требующие период дискретизации. Например: AP_FILTER, RAMPGEN и др.

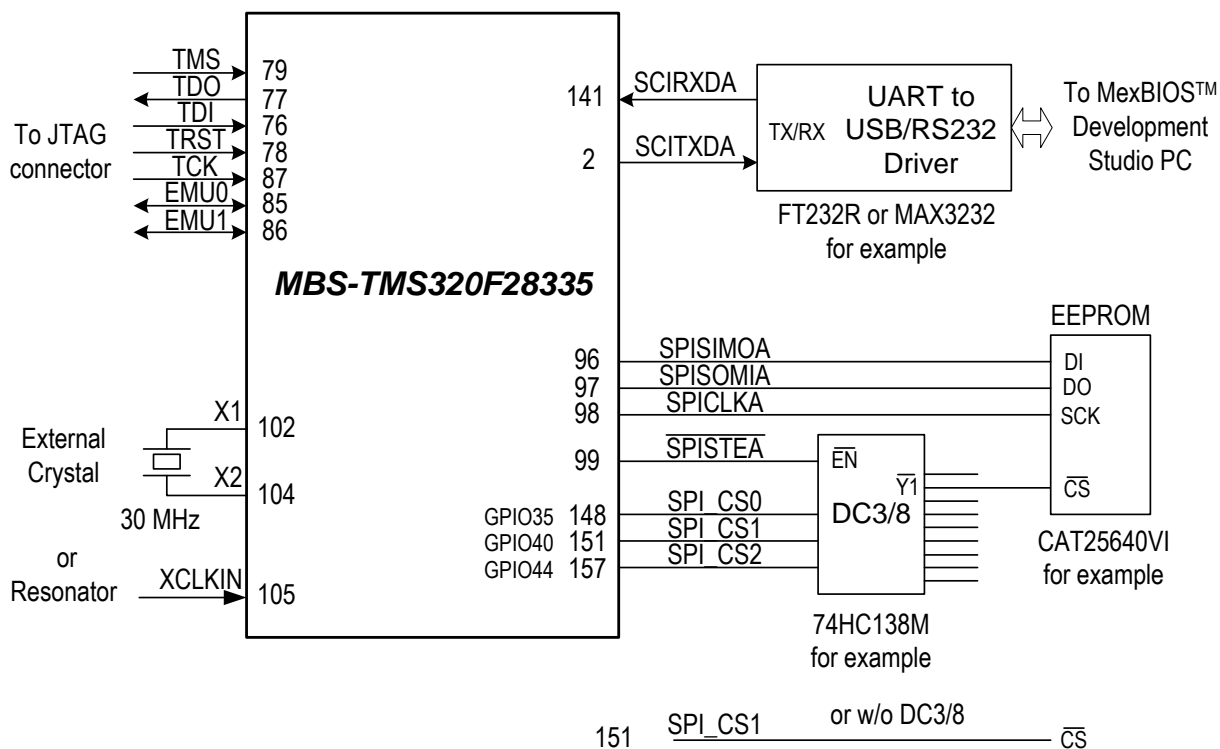
Если частота прерывания TINT0 будет изменена – необходимо учесть изменение во всех используемых схемах.

Описание стартового проекта

Проект «**MBS_start**» предназначен для конфигурации периферии микроконтроллера и запуска основных задач. Проект создан в среде Code Composer Studio 3.3.

Проект содержит следующие основные модули:

- «**DSP2833x_CodeStartBranch.asm**» – модуль, содержащий секцию точки входа в программу для ее запуска;
- «**DSP2833x_GlobalVariableDefs.c**» – модуль объявления периферийных регистров микроконтроллера;
- «**DSP2833x_Init.c**» – модуль конфигурации оборудования;
- «**main.c**» – главный модуль, содержащий секцию инициализации, фоновую задачу и обработчики прерываний;
- «**drv_comm_2833x.lib**» – библиотека драйвера коммуникации;
- «**config.h**» – модуль задания базовой конфигурации и определений.

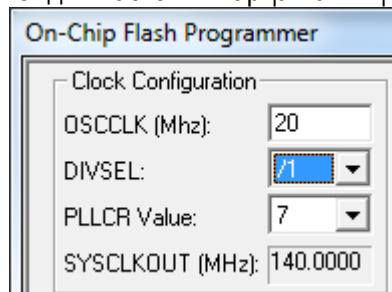


Модуль конфигурации оборудования «DSP2833x_Init.c»

Содержит следующие функции:

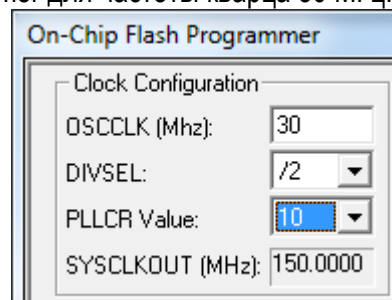
- **«InitSysCtrl»** – инициализация системных управляющих регистров (отключение сторожевого таймера, конфигурация PLL, разрешение работы встроенных модулей периферии);
- **«InitGpio»** – конфигурация ножек микроконтроллера (по умолчанию все задаются как входы);
- **«InitPieCtrl»** – конфигурация регистров управления менеджером прерываний;
- **«InitPieVectTable»** – инициализация таблицы векторов прерываний;
- **«InitXintf»** – инициализация внешнего интерфейса;
- **«InitEvRegs»** – конфигурация менеджеров событий (например, для работы PWM);
- **«InitCpuTimers»** – инициализация структур управления системными CPU таймерами;
- **«ConfigCpuTimer»** – конфигурация регистров заданного CPU таймера;
- **«MemCopy»** – функция копирования секции памяти;
- **«InitFlash»** – инициализация регистров управления FLASH.

Настройки On-Chip Flash Programmer для частоты кварца 20 МГц:



On-Chip Flash Programmer	
Clock Configuration	
OSCCLK (Mhz):	20
DIVSEL:	/1
PLLCR Value:	7
SYSCLKOUT (MHz):	140.0000

Настройки On-Chip Flash Programmer для частоты кварца 30 МГц:



On-Chip Flash Programmer	
Clock Configuration	
OSCCLK (Mhz):	30
DIVSEL:	/2
PLLCR Value:	10
SYSCLKOUT (MHz):	150.0000

Основной модуль «main.c»

Содержит следующие функции:

- **«main»** – базовая функция, управление на которую передается системой после секции старта кода.

В данной функции осуществляется вызов функций: конфигурации оборудования

(**InitHardware**), инициализация коммуникации (**SerialCommInit**), при наличии драйвера памяти функции чтения конфигурации операционной среды MexBIOS (**ReadMexBiosConfig**), функции задания конфигурации ядра ОС (**MBS_Create**) и функции разрешения работы прерываний.

Кроме того, осуществляется запуск фоновой задачи (бесконечного цикла), в которой осуществляется вызов функций установки конфигурации и запуска фоновых задач MexBIOS (**MBS_Init**, **MBS_IsrExecute**).

- **«CpuTimer0IsrHandler»** – обработчик прерывания для формирования основной частоты расчета (периода дискретизации системы).

В данной функции осуществляется вызов функций: функции реализации коммуникации по последовательному интерфейсу для загрузки конфигурации MexBIOS в микроконтроллер (**SerialCommTimings**), запуск периодических задач ОС (**MBS_IsrExecute**), записи конфигурации в энергонезависимую память при наличии команды (**WriteMexBiosConfig**), а также выполняется расчет загрузки системы и передача информации в ядро ОС (через **MbsCpuLoad**).

- **«SciaRxIsrHandler»** – обработчик прерывания по приему данных

В данной функции осуществляется вызов соответствующей функции из драйвера коммуникации.

- **«SciaTxIsrHandler»** – обработчик прерывания при передаче данных

В данной функции осуществляется вызов соответствующей функции из драйвера коммуникации.

Модуль конфигурации «config.h»

В данном модуле осуществляется подключение необходимых заголовочных файлов, а также определение системных констант:

- **«VERSION»** – версия проекта;
- **«SYSCLK»** – системная тактовая частота определяющая количество исполняемых операций в секунду;
- **«CLKIN»** – частота внешнего кварца (для определения множителя частоты);
- **«HZ»** – частота основного прерывания (для модуля CPU-Timer0).

Подключение драйвера коммуникации

Драйвер «drv_comm» предназначен для загрузки конфигурации MexBIOS в микроконтроллер. Для его подключения необходимо выполнить следующие этапы:

- 1) Подключение библиотеки драйвера к проекту

«Project» → «Add Files to Project ...» → выбрать из директории «lib» папки проекта библиотеку «drv_comm_2833x.lib»

- 2) Подключить в модуле «config» заголовочный файл драйвера

```
#include "drv_comm.h"           // Communication library header
```

- 3) Объявить прототипы функций обработки прерываний в модуле конфигурации «InitHardware»

```
// interrupt handler prototypes
interrupt void CpuTimer0IsrHandler (void);
interrupt void SciaRxIsrHandler (void);
interrupt void SciaTxIsrHandler (void);
interrupt void PIE_RESERVED (void);
```

- 4) Добавить их в таблицу векторов прерываний

```
// User interrupts that are used in this project are re-mapped to
// ISR functions found within this file
EALLOW;
PieVectTable.TINT0 = &CpuTimer0IsrHandler;
PieVectTable.RXAINT = &SciaRxIsrHandler;
PieVectTable.TXAINT = &SciaTxIsrHandler;
EDIS;
```

- 5) Разрешить прерывания по приему и передаче данных в менеджере обработки прерываний:

```
// Enable user interrupts
IER |= (M_INT1|M_INT9);

PieCtrlRegs.PIEIER1.bit.INTx7 = 1; // TINT0
PieCtrlRegs.PIEIER9.bit.INTx1 = 1; // SCIRXINTA
PieCtrlRegs.PIEIER9.bit.INTx2 = 1; // SCITXINTA
```

- 6) Выставить ножки микроконтроллера на работу функции в режиме SCI

```
EALLOW;

// User configuration
GpioCtrlRegs.GPAPUD.bit.GPIO28 = 0; // Enable pull-up for GPIO28 (SCIRXDA)
GpioCtrlRegs.GPAPUD.bit.GPIO29 = 0; // Enable pull-up for GPIO29 (SCITXDA)
GpioCtrlRegs.GPAQSEL2.bit.GPIO28 = 3; // Asynch input GPIO28 (SCIRXDA)
GpioCtrlRegs.GPAMUX2.bit.GPIO28 = 1; // Configure GPIO28 for SCIRXDA operation
GpioCtrlRegs.GPAMUX2.bit.GPIO29 = 1; // Configure GPIO29 for SCITXDA operation

EDIS;
```

- 7) В модуле «main.c» объявить структуру конфигурации и задать ее параметры:

```
// include base header file
#include "config.h"

// Communication parameters
TDrvCommParams CommParams = {
    0,           // UART identifier - SCIA
    1,           // Slave address
    1152,        // Baudrate / 100
    SYSCLK/8/115200-1, // Baudrate register value
    0,           // Parity mode - None
    5,           // Delay on byte reception (5/10000=0.0005 s)
    0,           // Delay before transmission begins (0 s)
    HZ/1000,     // Timing function frequency in kHz
    0,           // Transmission control function
#ifdef MBS_INCLUDE
    (TDrvCommCallback)MBS_GetDataAddr // Callback function
#else
    0
#endif
};
```



Примечание. В данном примере для обработки принятого кадра по последовательному интерфейсу используется функция ядра **MexBIOS (MBS_GetData)**. Пользователь при этом может создать собственную функцию и передать ее в драйвер коммуникации.

8) В функции «main» добавить инициализацию

```
// main function
void main(void)
{
    // hardware initialization
    InitHardware();

    // communication initialization
    SerialCommInit (&CommParams);
```

9) В обработчике основного системного прерывания добавить функцию реализации

```
interrupt void CpuTimer0IsrHandler(void)
{
    // clear timer counter for cpu load calculation
    CpuTimer1Regs.TIM.all = 0;

    // increment interrupt counter
    CpuTimer0.InterruptCount++;

    // execute function for communication
    SerialCommTimings();
```

10) Объявить обработчики прерываний по приему и передаче данных и вызвать в них соответствующие функции:


```
// SCIA receive interrupt handler
interrupt void SciaRxIsrHandler(void)
{
    // run appropriate function from communication driver
    SerialCommRxHandler();
    // acknowledge interrupt
    PieCtrlRegs.PIEACK.bit.ACK9 = 1;
}

// SCIA transmit interrupt handler
interrupt void SciaTxIsrHandler(void)
{
    // run appropriate function from communication driver
    SerialCommTxHandler();
    // acknowledge interrupt
    PieCtrlRegs.PIEACK.bit.ACK9 = 1;
}
```

Подключение ядра MexBIOS

Пользователь может подключить ядро MexBIOS к своему проекту. Для подключения необходимо выполнить следующие этапы:

- 1) Подключить в модуле «**config**» заголовочный файл

```
#include "MBS_Import.h" // MexBIOS library import header
```



Примечание. Данный файл автоматически генерируется в папке библиотеки блоков «\NPF Mechatronica-pro\MexBIOS Development Studio\Extend\TMS320F2835»

- 2) В свою структуру конфигурации коммуникации добавить адрес

```
// Communication parameters
TDrvCommParams CommParams = {
    0, // UART identifier - SCIA
    1, // Slave address
    1152, // Baudrate / 100
    SYSCLK/8/115200-1, // Baudrate register value
    0, // Parity mode - None
    5, // Delay on byte reception (5/10000=0.0005 s)
    0, // Delay before transmission begins (0 s)
    HZ/1000, // Timing function frequency in kHz
    0, // Transmission control function
    #ifdef MBS_INCLUDE
    (TDrvCommCallback)MBS_GetDataAddr // Callback function
    #else
    0
    #endif
};
```

- 3) Выполнить операцию чтения конфигурации из энергонезависимой памяти

```
// if memory drive enable execute function for
// reading MexBIOS configuration
#ifdef 0
ReadMexBiosConfig(Address, MbsMtxAddr, MbsMtxSize);
#endif
```



Примечание. Здесь приведен пример вызова такой функции. Параметры этой функции «**MbsMtxAddr**» и «**MbsMtxSize**» определены в заголовочном файле «**MBS_Import.h**», а параметр «**Address**» может назначить по своему усмотрению.

- 4) Вызвать в секции инициализации функцию «**MBS_Create**» и передать параметры в ядро MexBIOS. После этого подать команду на запуск (***MbsEnable**=1).

```
// if MexBIOS enable execute function for
// setting it base parameters
#ifdef MBS_INCLUDE
MBS_Create();
*MbsAppVersion = VERSION; // set project version to display in the studio
*MbsEnable = 1;           // command to startup
#endif
```

- 5) В фоновой задаче вызвать функции задания конфигурации и исполнения фоновых задач

```
// run forewer
while(1)
{
    // if MexBIOS enable execute functions for
    // setting configuration and execute tasks
    #ifdef MBS_INCLUDE
    MBS_Init();
    MBS_TaskExecute();
    #endif
}
```

- 6) В обработчике основного прерывания «**CpuTimer0IsrHandler**» вызвать обработчик периодических задач ядра

```
// execute function with interrupt source
MBS_IsrExecute();
```

- 7) По команде выполнить операцию записи конфигурации в энергонезависимую память

```
// if memory driver enable check write command
#ifdef 0
if (*MbsWriteCmd)
{
    // write MexBIOS configuration in memory
    WriteMexBiosConfig(Address, MbsMtxAddr, MbsMtxSize);
    *MbsWriteCmd = 0;
}
#endif
```



Примечание. Здесь приведен пример вызова такой функции. Параметры этой функции «MbsMtxAddr» и «MbsMtxSize» определены в заголовочном файле «MBS_Import.h», а параметр «Address» может назначить по своему усмотрению.

- 8) Выполнить расчет загрузки системы и передать значение в ядро для отображения в среде разработки

```
// CPU load calculation
*MbsCpuLoad = (CPUL_GAIN * ((- (Uint16) CpuTimer1Regs.TIM.all)) >> 21);
```

- 9) Выделить адресное пространство памяти программ и данных в командном файле для ядра MexBIOS

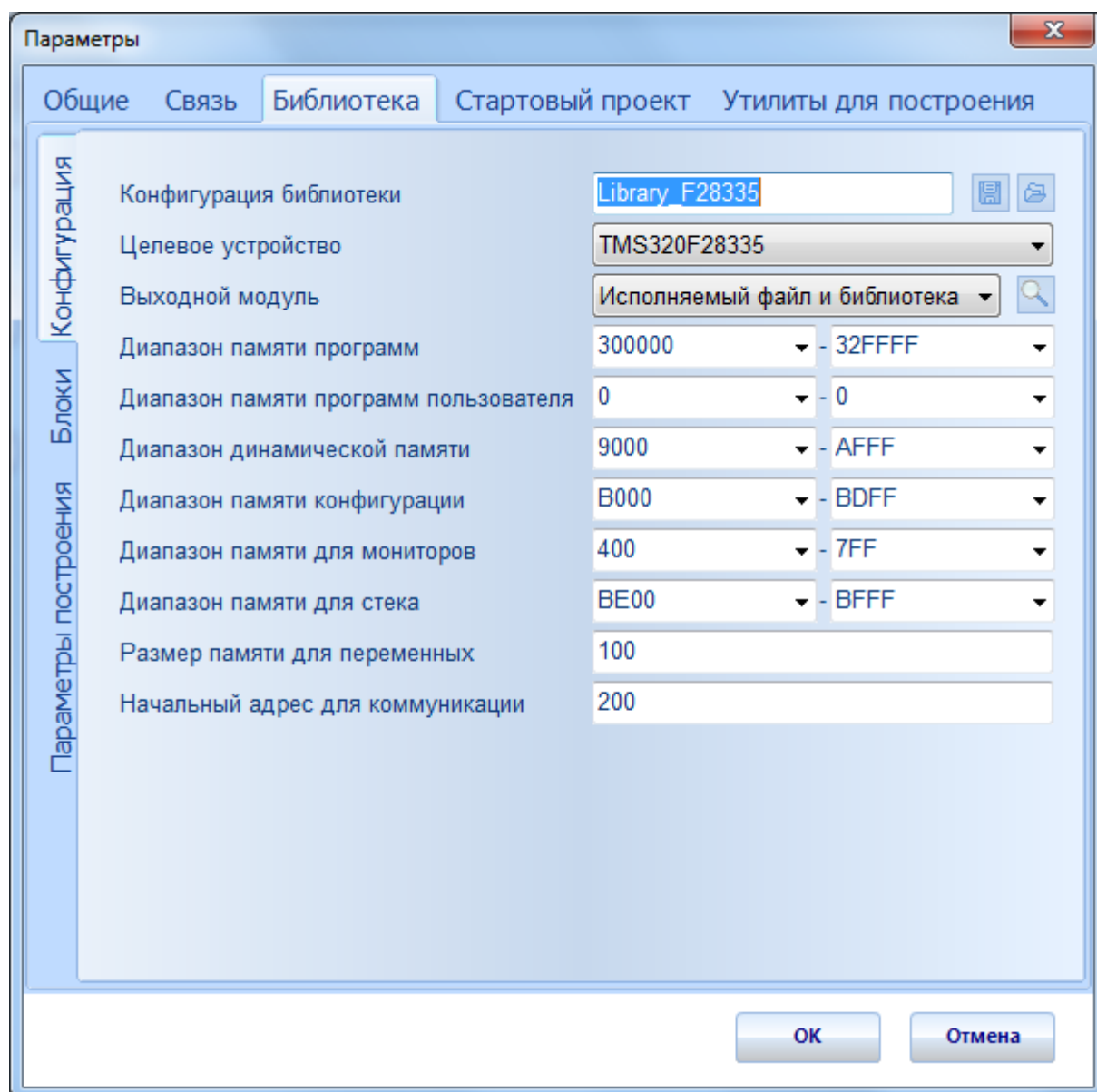
```
MEMORY
{
  PAGE 0:
    RAMM0      : origin = 0x000050, length = 0x0003B0
    ZONE0      : origin = 0x004000, length = 0x001000
    ZONE6      : origin = 0x100000, length = 0x100000
    ZONE7A     : origin = 0x200000, length = 0x00FC00
    FLASH      : origin = 0x330000, length = 0x00FF80
    /* FLASH   : origin = 0x300000, length = 0x030000
    CSM_RSVD   : origin = 0x33FF80, length = 0x000076
    BEGIN      : origin = 0x33FFF6, length = 0x000002
    CSM_PWL    : origin = 0x33FFF8, length = 0x000008
    OTP        : origin = 0x380400, length = 0x000400
    ADC_CAL    : origin = 0x380080, length = 0x000009
    IQTABLES   : origin = 0x3FE000, length = 0x000b50
    IQTABLES2  : origin = 0x3FEB50, length = 0x00008c
    FPUTABLES  : origin = 0x3FEBDC, length = 0x0006A0
    ROM        : origin = 0x3FF27C, length = 0x000D44
    RESET      : origin = 0x3FFFC0, length = 0x000002
    VECTORS     : origin = 0x3FFFC2, length = 0x00003E

  PAGE 1 : /* Data Memory */
    BOOT_RSVD  : origin = 0x000000, length = 0x000050
    /* RAMM1    : origin = 0x000400, length = 0x000400
    RAML0      : origin = 0x008000, length = 0x001000
    /* RAML13   : origin = 0x009000, length = 0x003000
    RAML4      : origin = 0x00C000, length = 0x001000
    RAML5      : origin = 0x00D000, length = 0x001000
    RAML6      : origin = 0x00E000, length = 0x001000
    RAML7      : origin = 0x00F000, length = 0x001000
    ZONE7B     : origin = 0x20FC00, length = 0x000400
}
```

- 10) Провести настройку памяти в MexBIOS™ Development Studio согласно произведённым изменениям.

Настройка памяти в MexBIOS

Данные настройки необходимо выставлять в соответствии с настройками, выставленными в окне **Параметры** вкладка **Правка** в главном меню, кнопка **Настройка**. На рисунке представлена настройка по умолчанию. Пользователь может создать свой файл настройки и произвести его сохранение. Сохранить и загрузить файл можно только из папки **Config** в корне программы.



Параметры	Значения
Конфигурация библиотеки	Library_F28335
Целевое устройство	TMS320F28335
Выходной модуль	Исполняемый файл и библиотека
Диапазон памяти программ	300000 - 32FFFF
Диапазон памяти программ пользователя	0 - 0
Диапазон динамической памяти	9000 - AFFF
Диапазон памяти конфигурации	B000 - BDFF
Диапазон памяти для мониторов	400 - 7FF
Диапазон памяти для стека	BE00 - BFFF
Размер памяти для переменных	100
Начальный адрес для коммуникации	200

Распределение Flash-памяти:

- для библиотеки и ядра MexBIOS
- для стартового проекта

– 0x300000 ÷ 0x32FFFF

– 0x330000 ÷ 0x33FF7F

Распределение RAM-памяти:

- для библиотеки и ядра MexBIOS
- для стартового проекта

– 0x9000 ÷ 0xBFFF, 0x000400 ÷ 0x0007FF

– 0x008000 ÷ 0x008FFF, 0x00C000 ÷ 0x00FFFF

Прошивка в контроллер MexBIOS kernel в плату с процессором TMS320F28335

В данном разделе описано несколько различных способов загрузки стартового проекта и библиотеки блоков с помощью различных аппаратных и программных средств.

В зависимости от наличия средств программирования далее необходимо выбрать один из способов загрузки стартового проекта и библиотеки блоков в flash память контроллера.

После загрузки библиотеки блоков и стартового проекта одним из способов можно переходить к непосредственному программированию контроллера с помощью MexBIOS Development Studio.

Компиляция библиотеки блоков и стартового проекта

Компиляция библиотеки требуется, если было произведено изменение адресного пространства для расположения стартового проекта или библиотеки блоков, добавление нового или редактирование существующего блока в библиотеке TMS320F2833x. Также если выбран другой контроллер или отладочная плата другого производителя. После компиляции необходимо загрузить стартовый проект и библиотеку блоков в контроллер заново.

Если никаких изменений не производилось, то нет необходимости компилировать библиотеку.

1. Если требуется компиляция библиотеки блоков и стартового проекта – установите компилятор от Texas Instruments. Code Generation Tools можно найти и скачать по ссылке https://www.ti.com/downloads/sds_support/TICodegenerationTools/download.htm (нужна регистрация на ti.com).

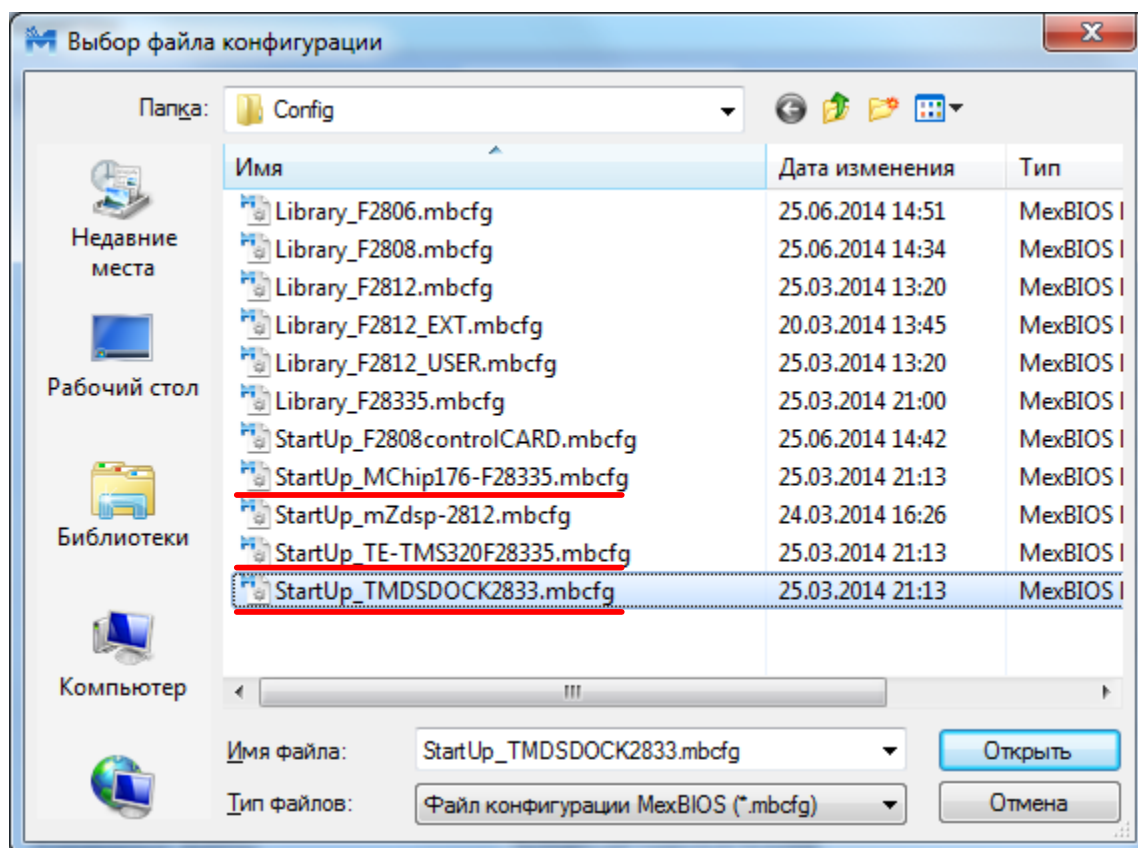
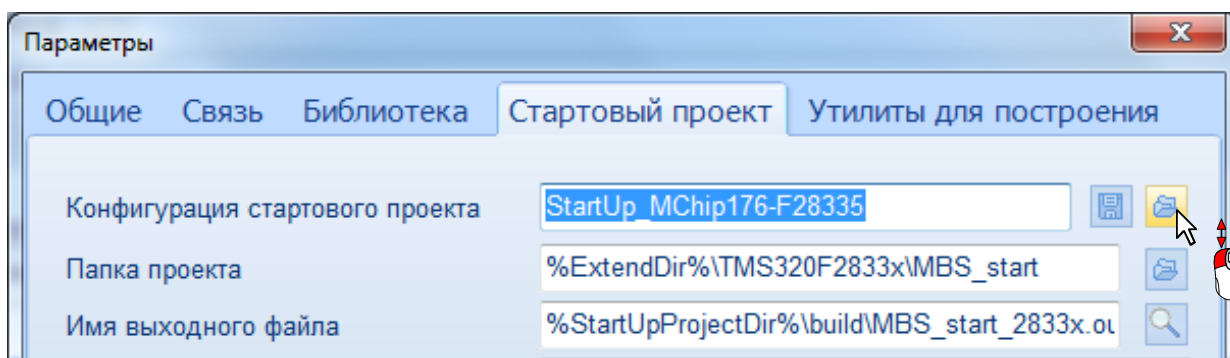
1.1. После установки компилятора открыть **MexBIOS Development Studio**.

1.2. Создать файл проекта для **TMS320F2833x**. Вкладка **Домой**, кнопка **Создать** → **TMS320F2833x**.

1.3. Перейти на вкладку **Правка**. Нажать кнопку **Построить**. Произойдёт компиляция библиотеки Models, TMS320F2833x и стартового проекта.

1.4. Если компиляции не прошла, убедитесь, что путь к компилятору задан верно. Для этого на вкладке **Правка** нажать клавишу **Настройки**. На вкладке **Утилиты для построения** в строке **Папка Code Generation Tool** должен быть указан путь к установленному компилятору. Необходимо использовать версию компилятора не ниже 5.2.1

2. MexBIOS Development Studio поддерживает работу с отладочными комплектами MChip176-TMS320F2833x, TE-TMS320F28335, TMDSDOCK2833. Для работы с одной из плат необходимо в окне **Параметры**, на вкладке **Стартовый проект** в опции **Конфигурация стартового проекта** выбрать нужный файл конфигурации:



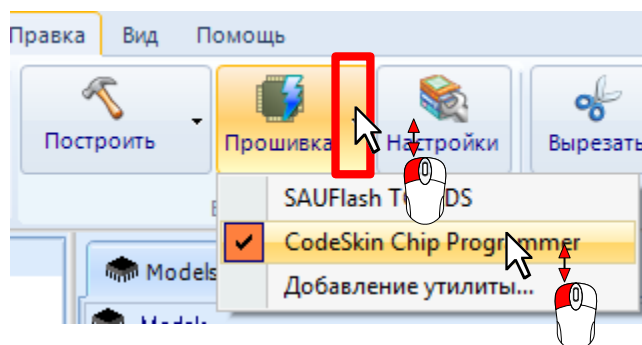
3. Нажать ОК и закрыть окно настроек.

4. На вкладке **Правка** главного меню, нажать кнопку **Построить**. После компиляции можно приступить к загрузке.

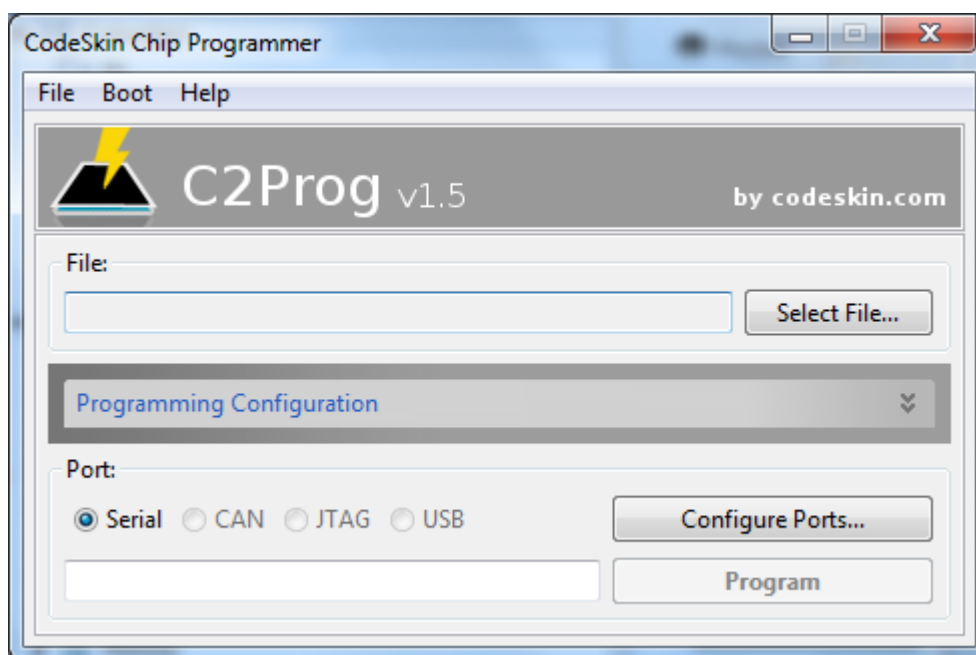
Загрузка с помощью C2Prog и эмулятора XDS100v1,v2

Загрузка возможна только эмулятором XDS100 v1, v2.

1. Загрузите и установите C2Prog с сайта производителя. (<http://www.codeskin.com/programmer>).
2. Настройте вызов программы для загрузки.
 - 2.1. Перейти на вкладку **Правка** в MexBIOS™ Development Studio.
 - 2.2. Нажать на маленький треугольник кнопки **Прошивка**.

Запуск **C2Prog**

- 2.3. Нажать на пункт **CodeSkin Chip Programmer**.
- 2.4. В появившемся окне нажать кнопку **Выбрать**. Окно настроек закроется.
3. Нажать непосредственно на кнопку **Прошивка**.
4. Появится окно **C2Prog**. Вид окна при первом запуске представлен на следующем рисунке:

Внешний вид **C2Prog** при первом запуске

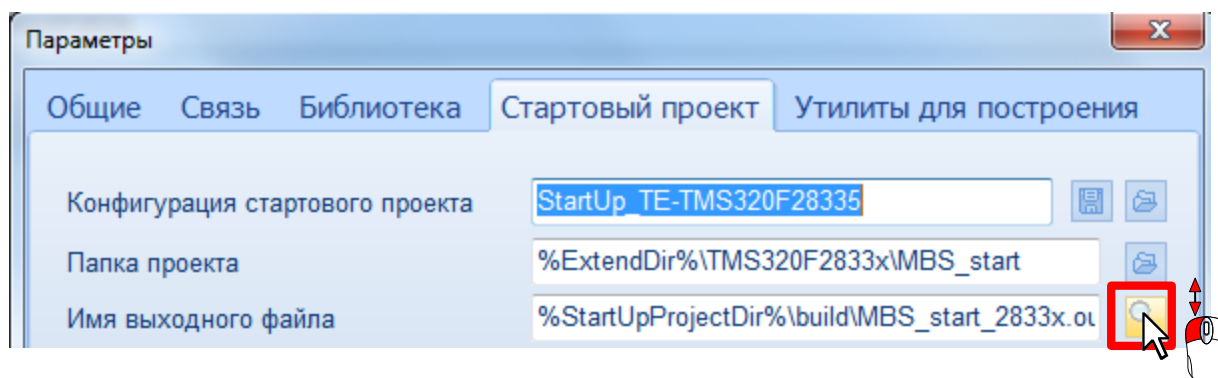
Если окно C2Prog не появилось, убедитесь, что программа установлена.

5. Нажать на кнопку **Select File...** В появившемся диалоговом окне выбрать сгенерированный **.hex** файл. Нажать **Open**.

Примечание. Для библиотеки процессоров серии **2833x** загружаемый **.hex** файл стартового проекта расположен по адресу:

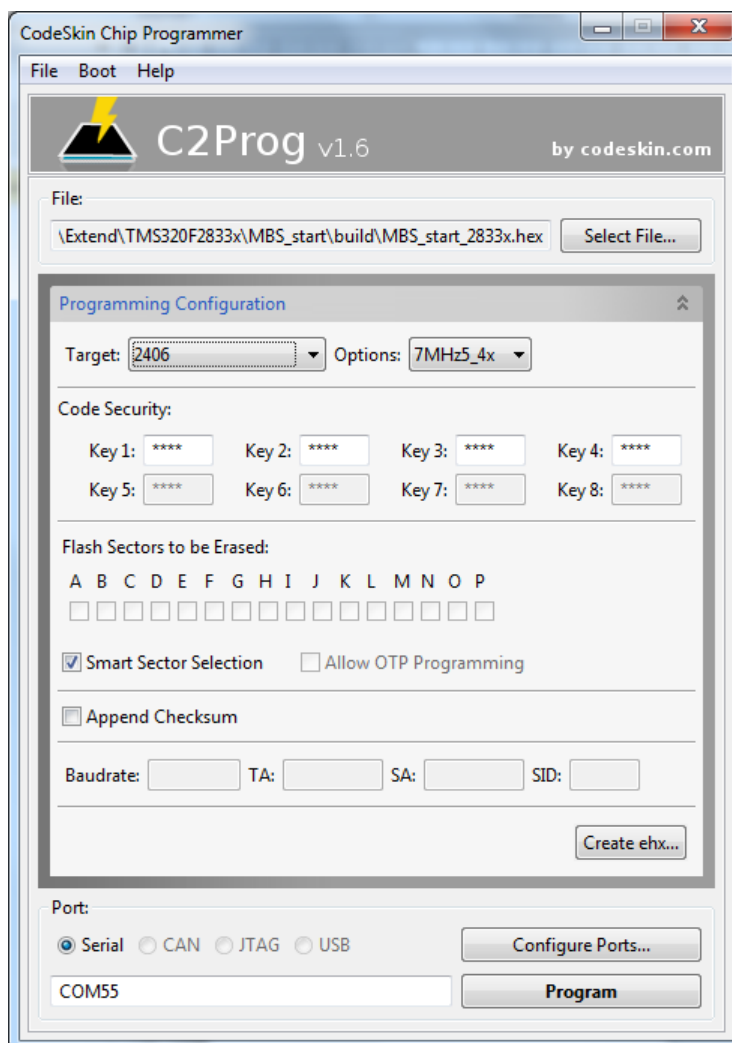
C:\Users\%UserName%\AppData\Roaming\NPF Mechatronica-Pro\MexBIOS Development Studio\Libraries\TMS320F2833x\StartUp\build

Папку стартового проекта можно открыть через окно **Параметры** в MexBIOS Development Studio:



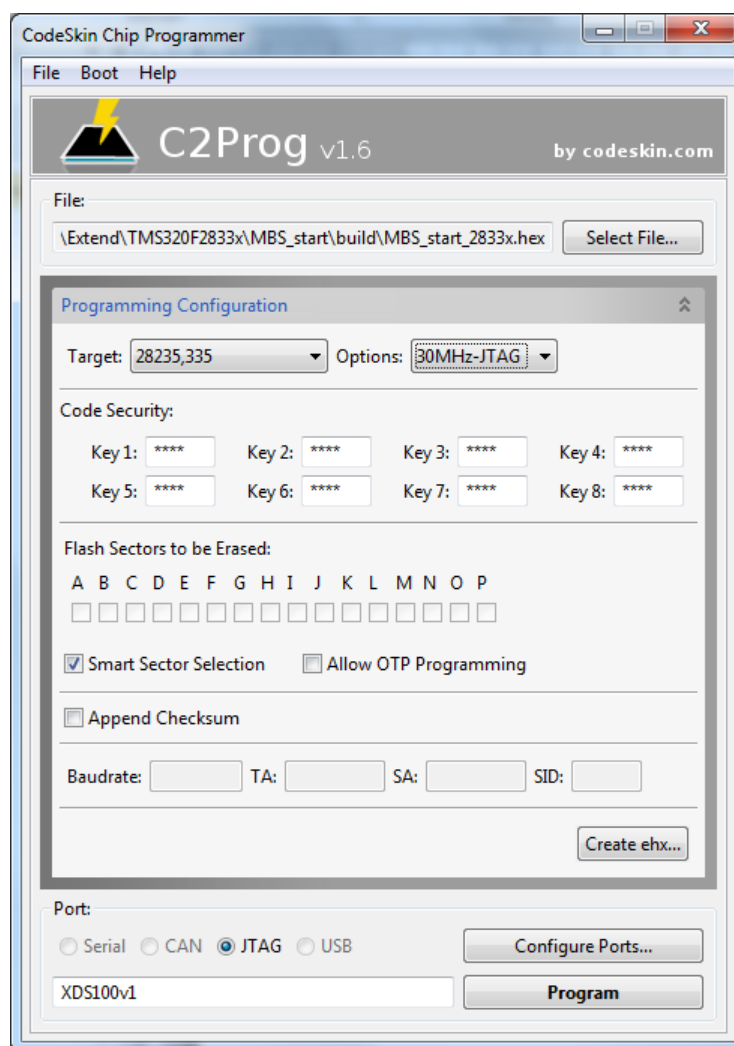
Открытие папки с Out файлами стартового проекта

6. После выбора .hex файла стартового проекта главное окно **C2Prog** изменится:



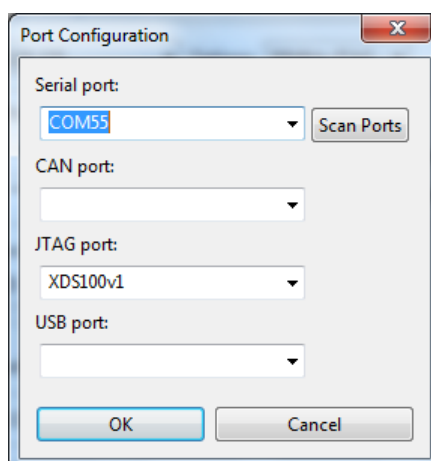
Промежуточный вид окна C2Prog

7. Далее необходимо в списке **Target** выбрать процессор **28235,335**.

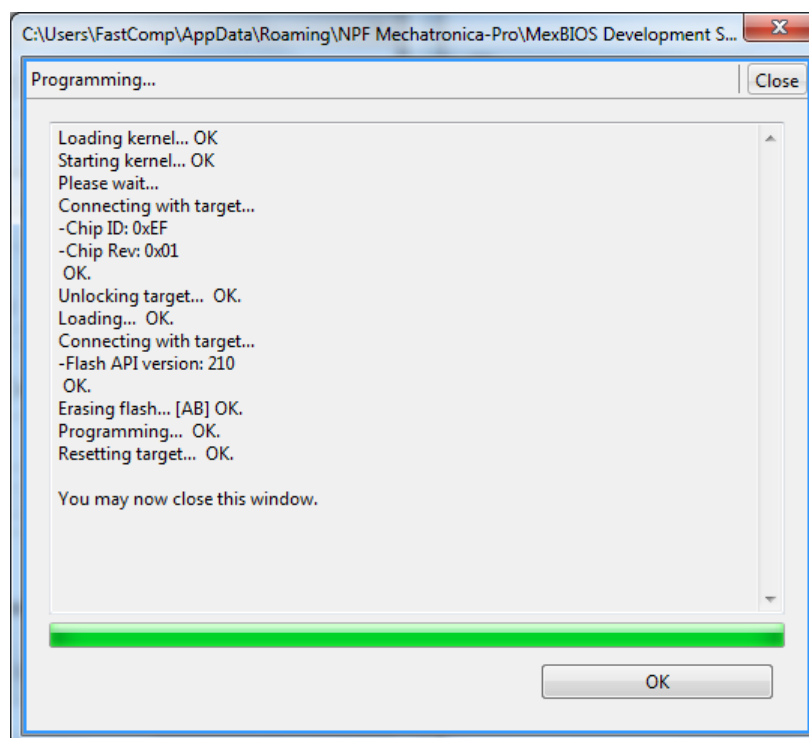


8. После выбора процессора в списке **Options** необходимо выбрать **30MHz-JTAG**.
9. В настройках **Configure Ports...** выбрать **COM** порт, к которому подключен JTAG эмулятор XDS100. Смотрите следующий рисунок.
 - 9.1. Если порт не появился – нажать кнопку Scan Ports.

Примечание: возможно загрузка не будет происходить, если к плате подключен и эмулятор и мини USB. В случае сбоя загрузки необходимо сбросить питание с платы и эмулятора. Затем подключить и попробовать снова.



10. Закрыть диалог настройки порта.
11. Нажать кнопку **Program**. Начнётся процесс загрузки:

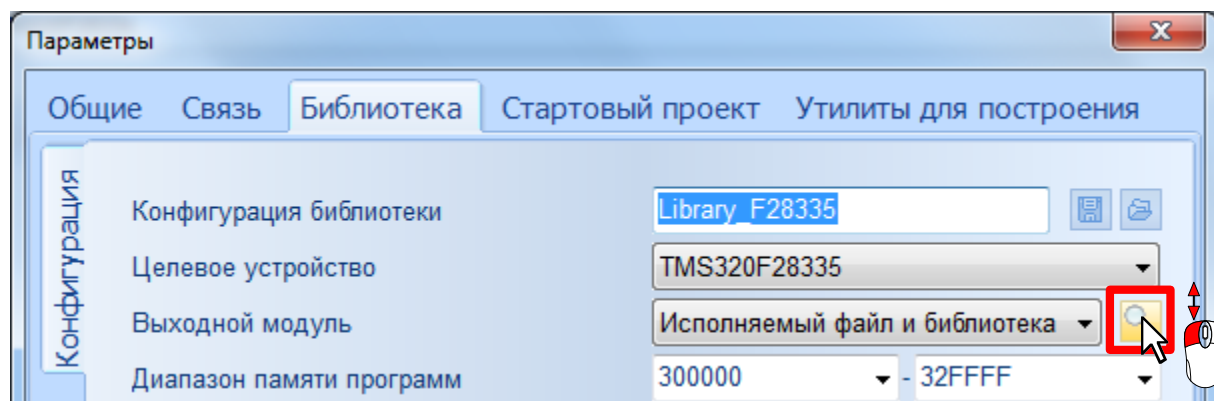


12. После успешной загрузки стартового проекта, нажать на кнопку **Select File...** для загрузки библиотеки блоков. В появившемся диалоговом окне выбрать сгенерированный **.hex** файл библиотеки блоков.

Примечание. Загружаемый **.hex** файл библиотеки блоков расположен по адресу:

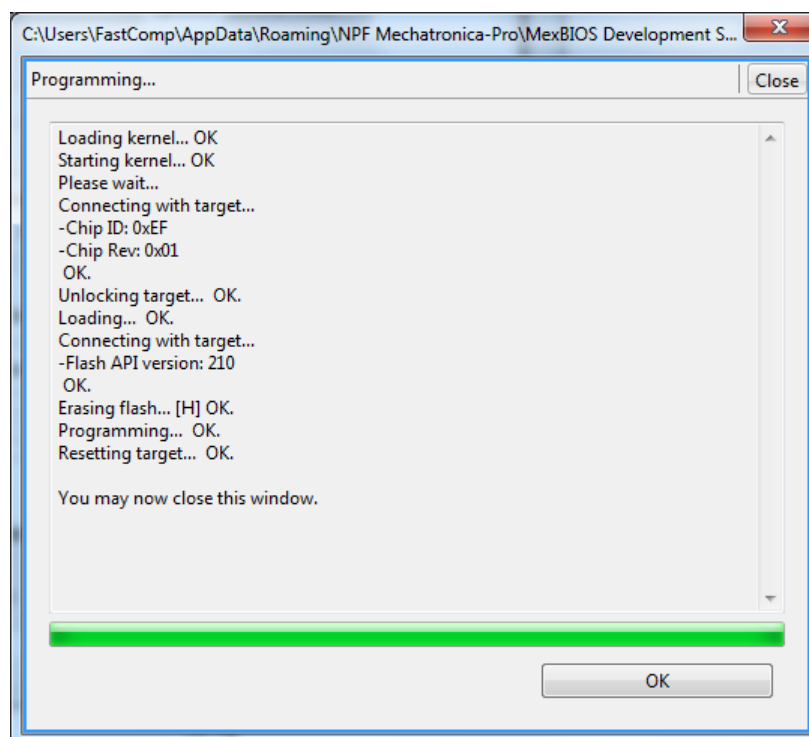
C:\Users\%UserName%\AppData\Roaming\NPFI Mechatronica-Pro\MexBIOS Development Studio\Libraries\TMS320F2833x\TMS320F2833x.hex

Папку библиотеки проекта можно открыть через окно **Параметры** в MexBIOS Development Studio:



Открытие папки с Out файлами библиотеки в MexBIOS Development Studio

13. Нажать кнопку **Program**.

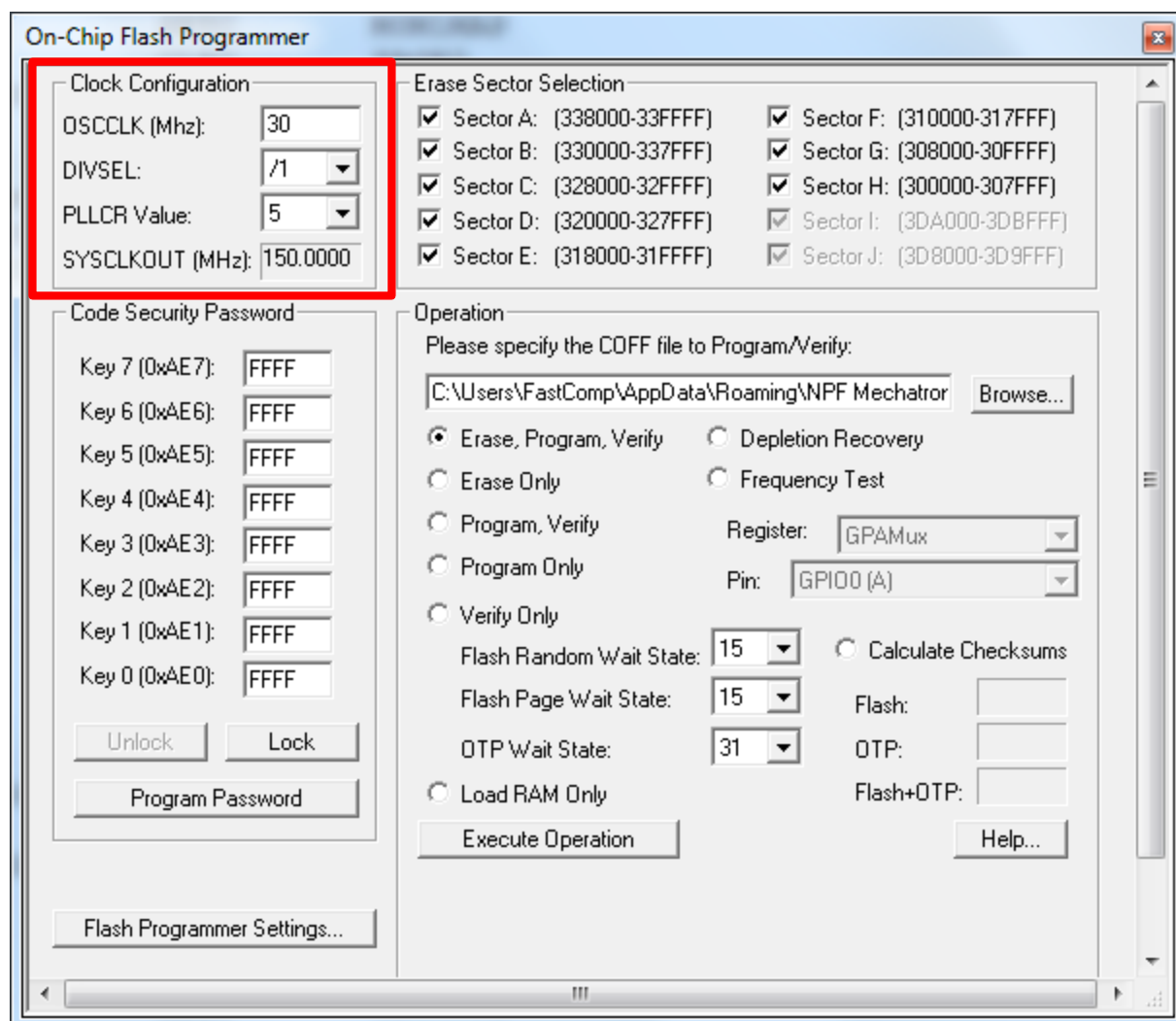


Стартовый проект и библиотека блоков загружены в память контроллера. Устройство готово к работе с MexBIOS Development Studio. Необходимо сбросить питание платы и отключить эмулятор и подключить Mini-Usb кабель к плате.

Ядро MexBIOS установлено. Аппаратное обеспечение готово для совместной работы с MexBIOS Development Studio.

Загрузка с помощью Code Composer v3.3 и эмулятора XDS100 v1,v2

1. Отключить от платы miniUSB. Подключить эмулятор XDS100, подключить питание +5 В.
2. Открыть Code Composer Studio v3.3.
 - 2.1. При необходимости добавить в Setup CCStudio v3.3 эмулятор XDS100 Usb. Нажать кнопку **Save&Quit**. Откроется окно Code Composer Studio.
3. Подключиться к плате, нажав **Alt+C**.
4. В окне Code Composer Studio, в меню **Tools** открыть **28xx On-Chip Flash Programmer**.
5. Если появилось окно промежуточных настроек, то смотрите настройки **Clock Configuration** на следующем рисунке.



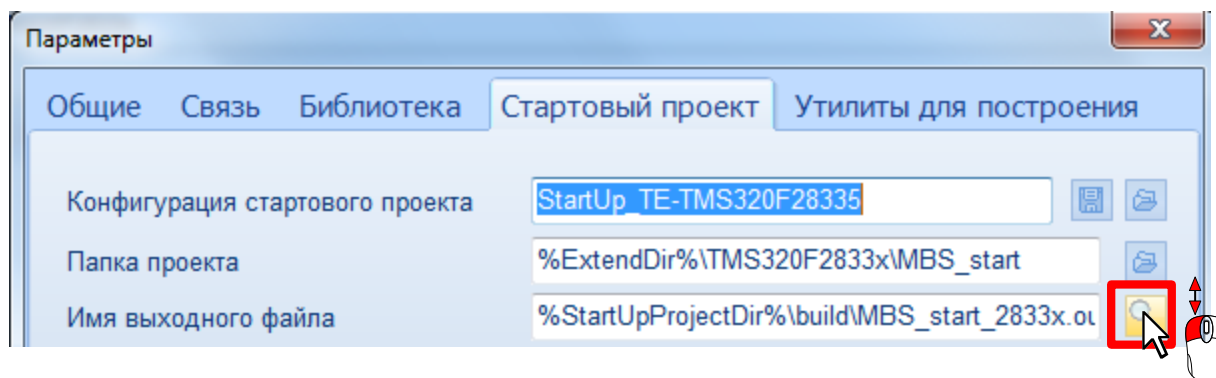
Параметры **OSCCLK**, **DIVSEL**, **PLLCR Value** установить, как показано на рисунке. Для загрузки во flash память также можно установить параметры **OSCCLK=30**, **DIVSEL=2**, **PLLCR=10 Value**.

5.1. Выбрать MBS_start.out файл.

Примечание. Для библиотеки процессоров серии **2833x** загружаемый **.out** файл стартового проекта расположен по адресу:

...\\MexBIOS Development Studio\\Extend\\TMS320F2833x\\MBS_start\\build\\MBS_start.out

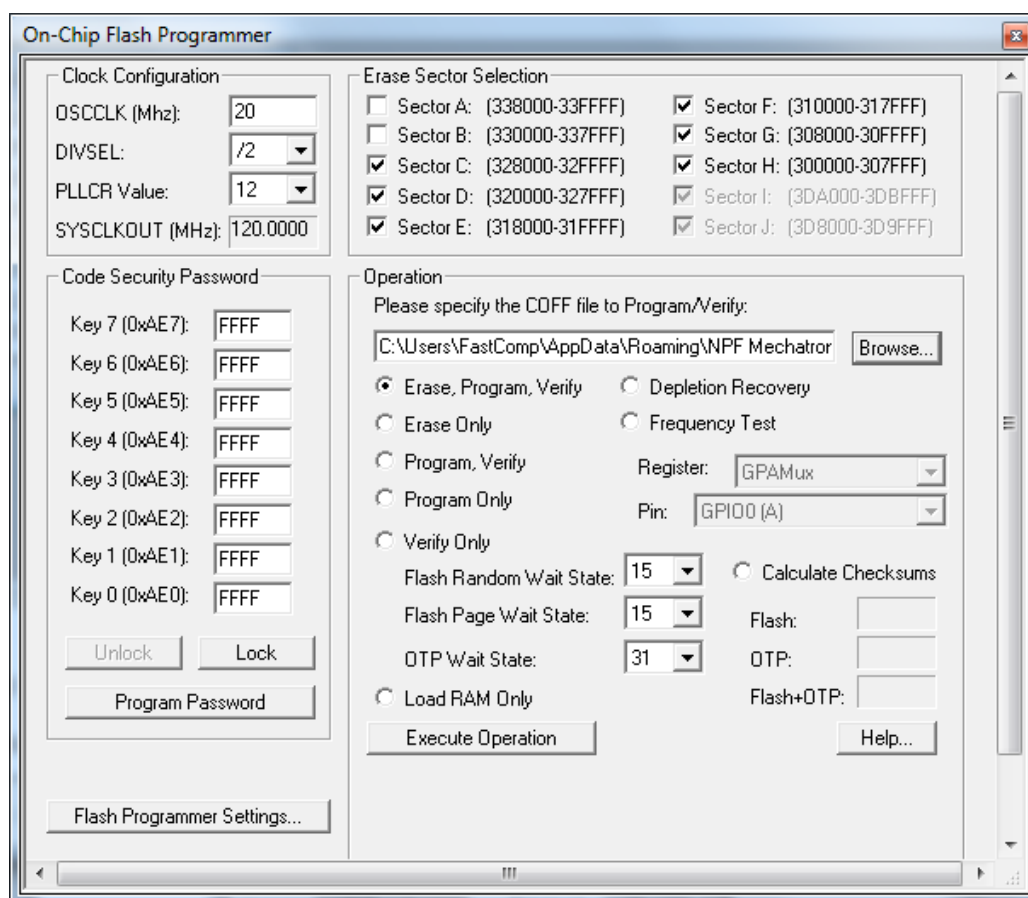
Папку стартового проекта можно открыть через окно **Параметры** в MexBIOS Development Studio:



Открытие папки с Out файлами стартового проекта для копирования адреса расположения

5.2. После выбора out файла нажать кнопку **Execute Operation**. Произойдёт запись стартового проекта в выбранные сектора.

6. Для загрузки библиотеки блоков выбрать следующие сектора:

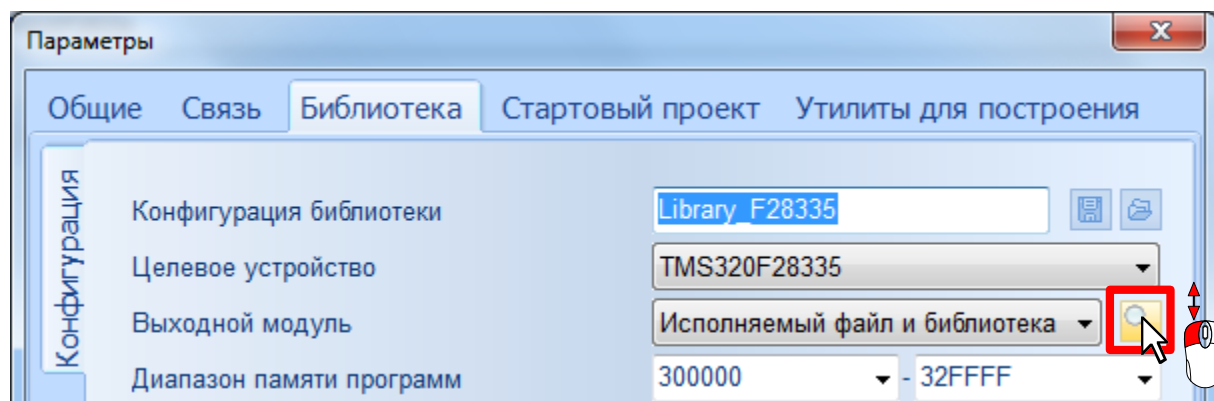


Параметры **OSCCLK**, **DIVSEL**, **PLLCR Value** установить, как показано на рисунке. Для загрузки во flash память также можно установить параметры **OSCCLK=30**, **DIVSEL=2**, **PLLCR=10 Value**.

6.1. Нажать кнопку **Browse...**

Примечание. Загружаемый **.out** файл библиотеки блоков расположен по адресу:
...\\MexBIOS Development Studio\\Extend\\TMS320F2833x\\TMS320F2833x.out

Папку библиотеки проекта можно открыть через окно **Параметры** в MexBIOS Development Studio:



Открытие папки с Out файлами библиотеки для копирования адреса расположения

6.2. Выбрать файл **TMS320F2833x.out**.

6.3. Нажать кнопку **Execute Operation**. Произойдёт запись библиотеки блоков в выбранные сектора.

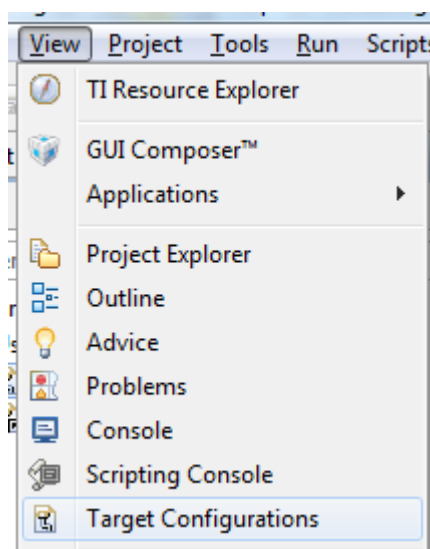
Ядро MexBIOS и библиотека блоков установлены. Аппаратное обеспечение готово для совместной работы с MexBIOS Development Studio.

Загрузка стартового проекта и библиотеки блоков с помощью Code Composer Studio 5.4 и эмулятора XDS100v3

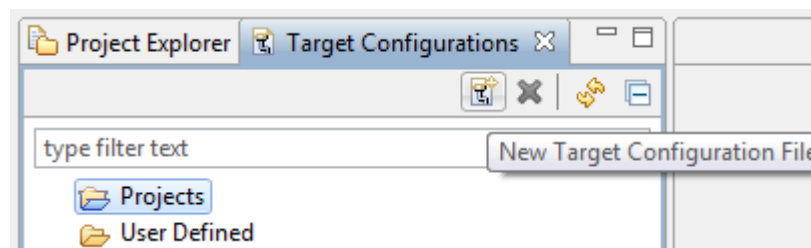
Далее будет показана загрузка стартового проекта и библиотеки блоков с помощью Code Composer v 5.4 и эмулятора XDS100v3.

Примечание: По следующей инструкции можно загрузить стартовый проект и библиотеку с помощью Code Composer v4.xx, с использованием эмулятора XDS100 v1 или v2. Для этого в конфигурации нужно выбрать соответствующий эмулятор, остальные действия аналогичны описанным.

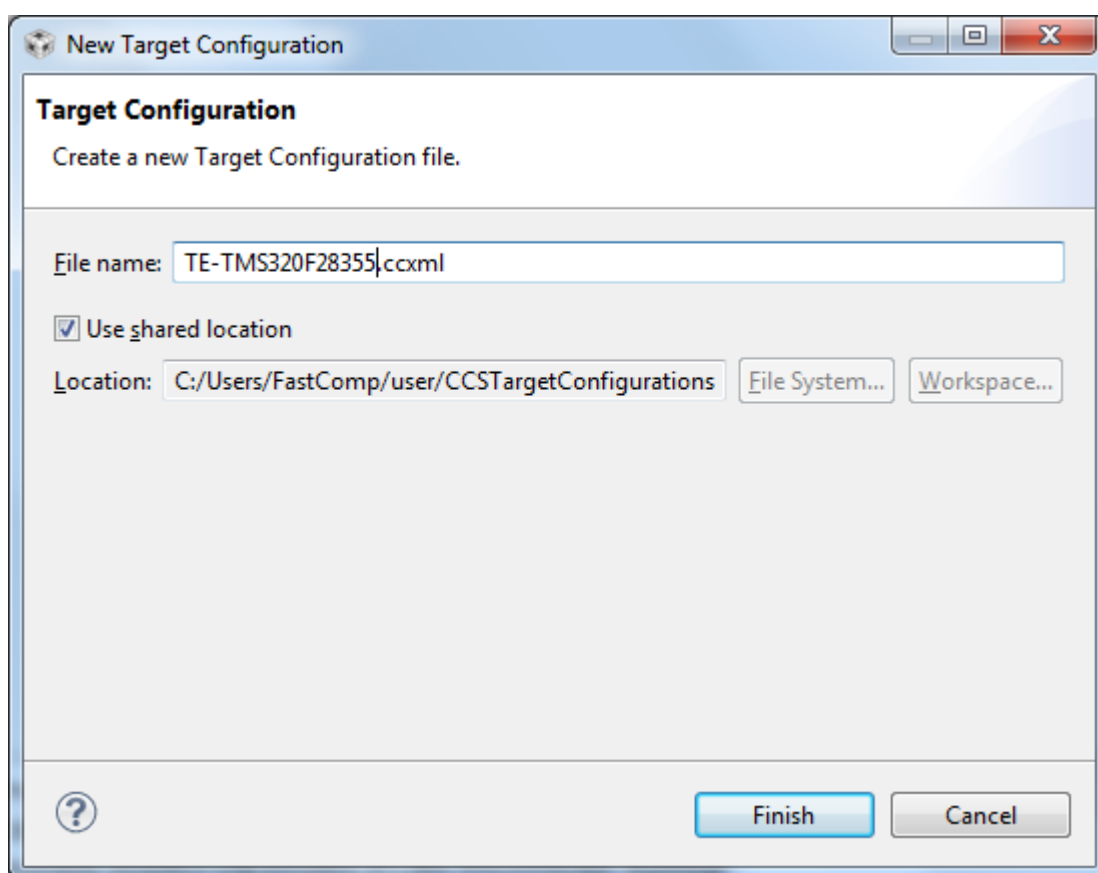
1. Запустите CCS.
2. Необходимо создать файл конфигурации с необходимыми настройками:
 - 2.1. Для открытия окна конфигураций **View**→**Target Configurations** (см. следующий рисунок)



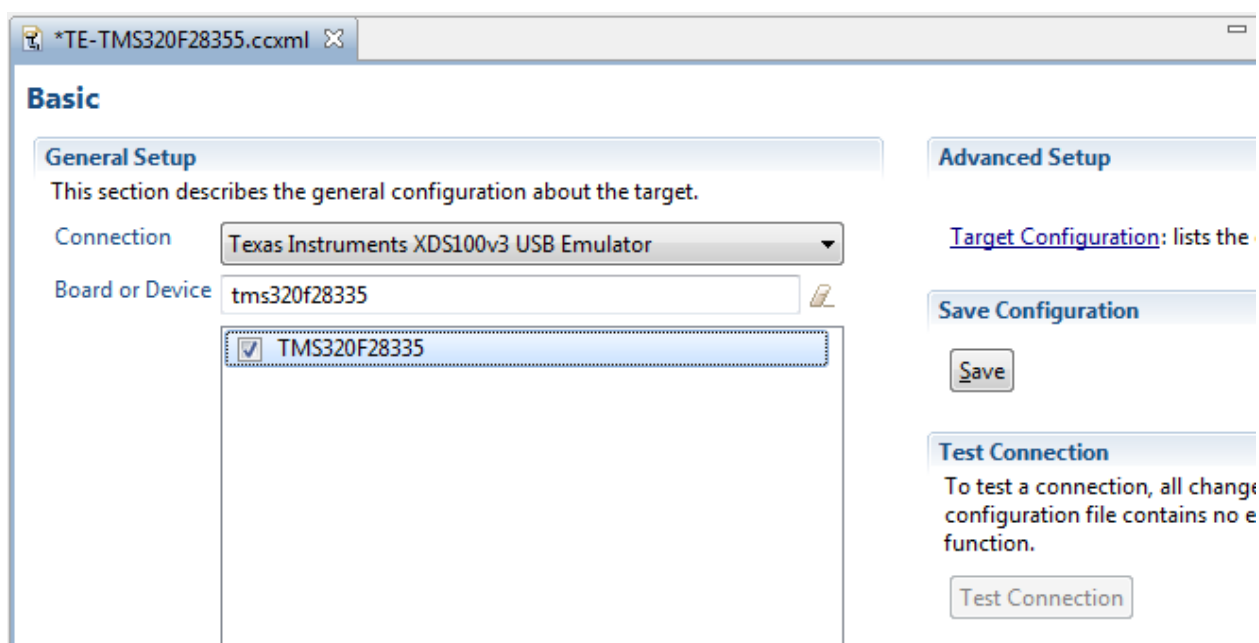
- 2.2. Нажать кнопку добавления конфигурации (см. следующий рисунок)



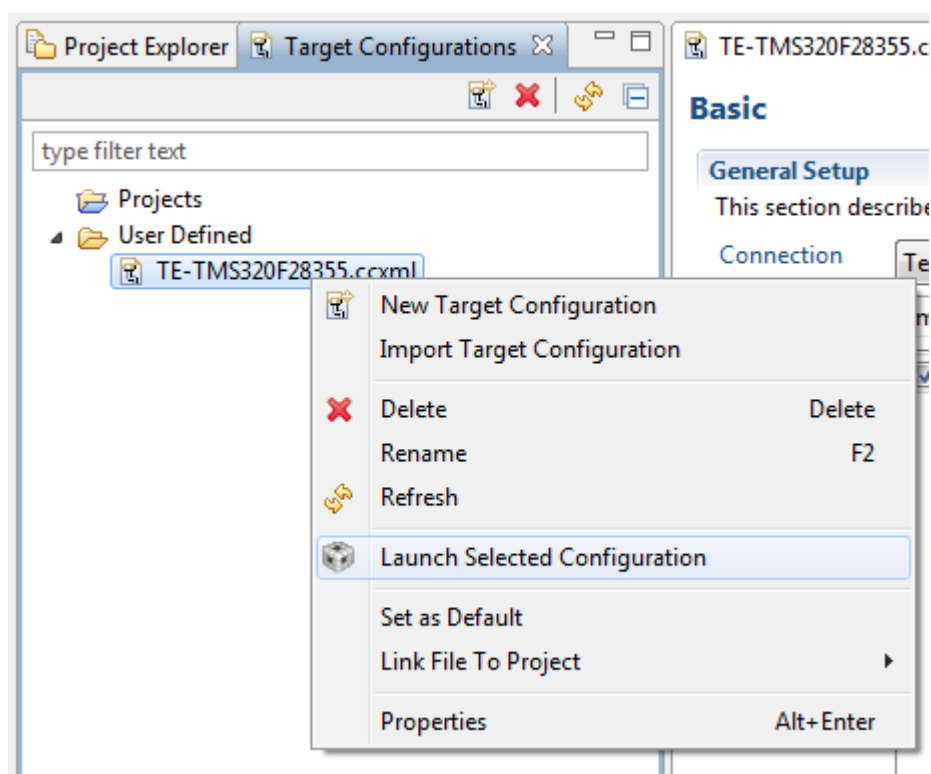
- 2.3. Назвать файл конфигурации (см. следующий рисунок)



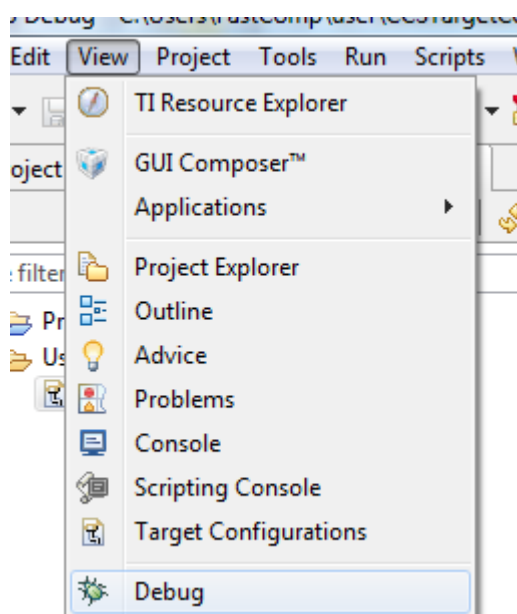
- 2.4. В появившемся окне конфигурации TE-TMS320F28355.ccxml выбрать способ подключения Texas Instruments XDS100v3 USB Emulator. В списке Board or Device набрать название процессора. Нажать кнопку Save.



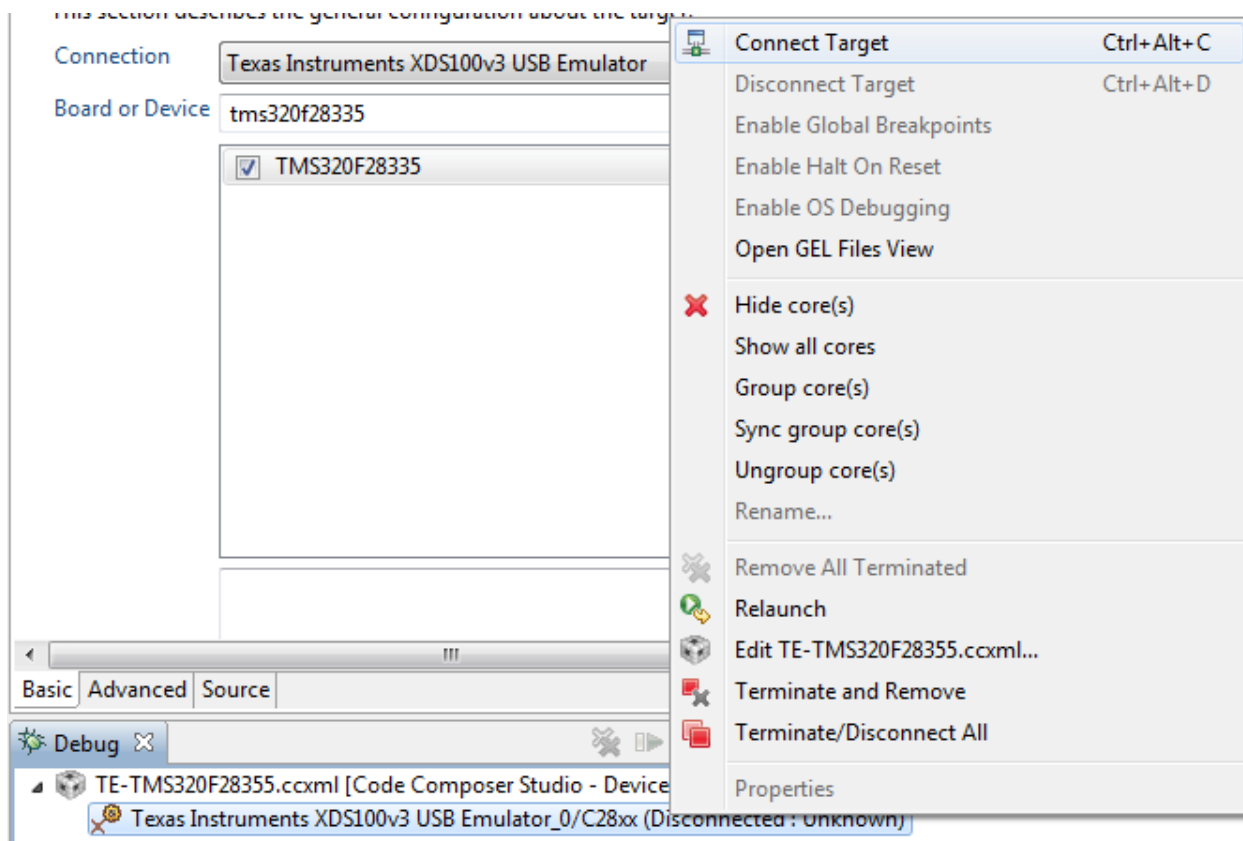
3. В окне **Target Configurations** нажать правой кнопкой мыши по файлу конфигурации и выбрать **Launch Selected Configuration**.



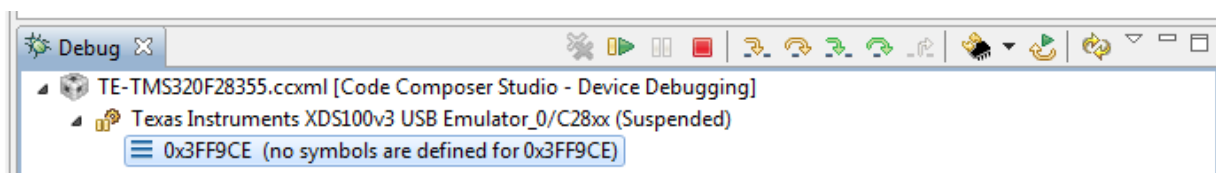
3.1. Если диалог **Debug** не открыт, то его можно открыть:



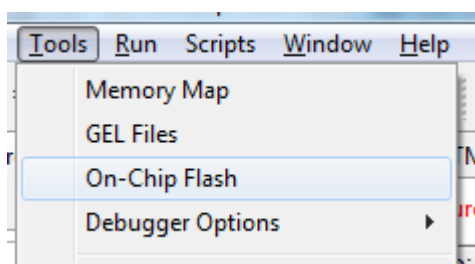
4. В окне **Debug** нажать правой кнопкой мыши по эмулятору и выбрать в появившемся меню пункт **Connect Target**.



Если эмулятор подключен к плате и на плату подано питание произойдет подключение к плате. На следующем рисунке показано изменение окна Debug при установленной связи.

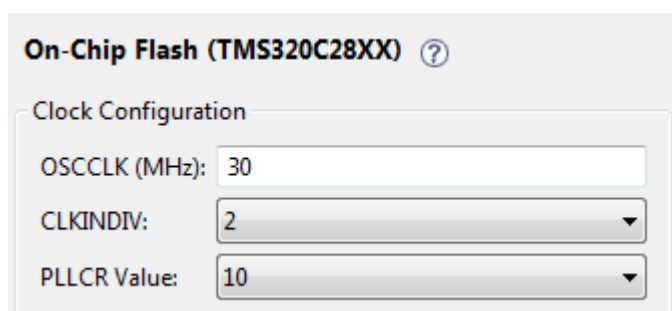


5. Открыть окно **On-Chip Flash**.



6. В появившемся окне **On-Chip Flash** произвести следующие настройки:

6.1. Настроить **Clock Configuration**. На плате установлен кварц 20 MHz, но частота, на которой процессор производит расчёты 150 MHz, поэтому загрузка выполниться со следующими настройками.



On-Chip Flash (TMS320C28XX) ?

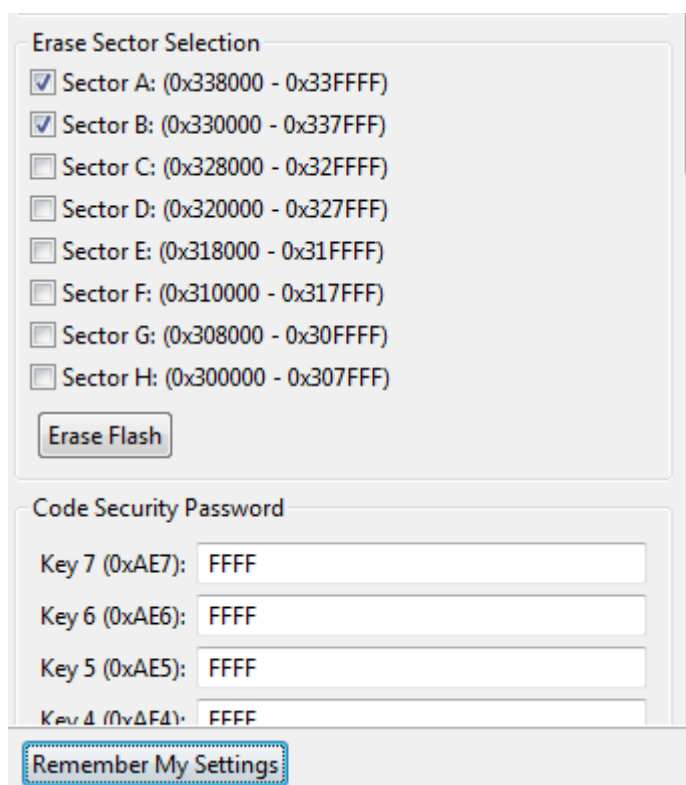
Clock Configuration

OSCCLK (MHz): 30

CLKINDIV: 2

PLLCR Value: 10

6.2. Для загрузки стартового проекта оставить выделенными сектора А и В. Обязательно нажать кнопку **Remember My Settings**.



Erase Sector Selection

☒ Sector A: (0x338000 - 0x33FFFF)

☒ Sector B: (0x330000 - 0x337FFF)

☐ Sector C: (0x328000 - 0x32FFFF)

☐ Sector D: (0x320000 - 0x327FFF)

☐ Sector E: (0x318000 - 0x31FFFF)

☐ Sector F: (0x310000 - 0x317FFF)

☐ Sector G: (0x308000 - 0x30FFFF)

☐ Sector H: (0x300000 - 0x307FFF)

Erase Flash

Code Security Password

Key 7 (0xAE7): FFFF

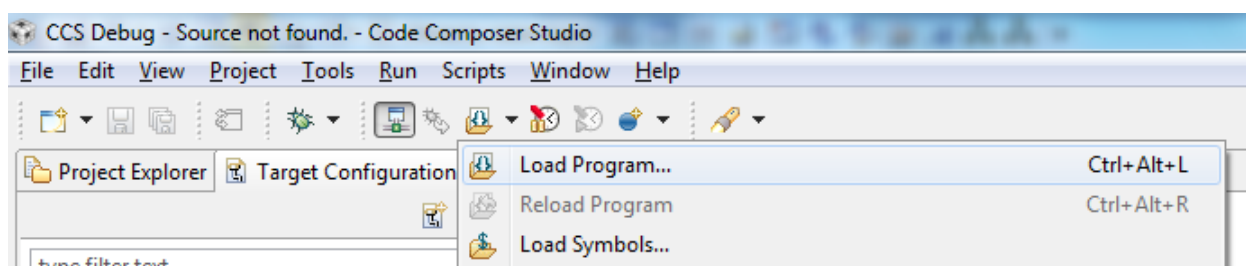
Key 6 (0xAE6): FFFF

Key 5 (0xAE5): FFFF

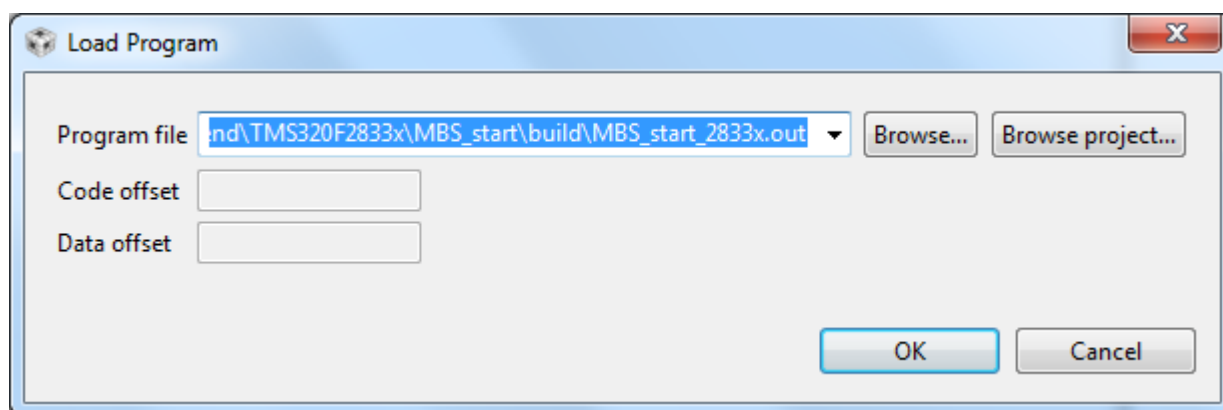
Key 4 (0xAE4): FFFF

Remember My Settings

7. Нажать на кнопку **Load** или выбрать команду **Load Program...**



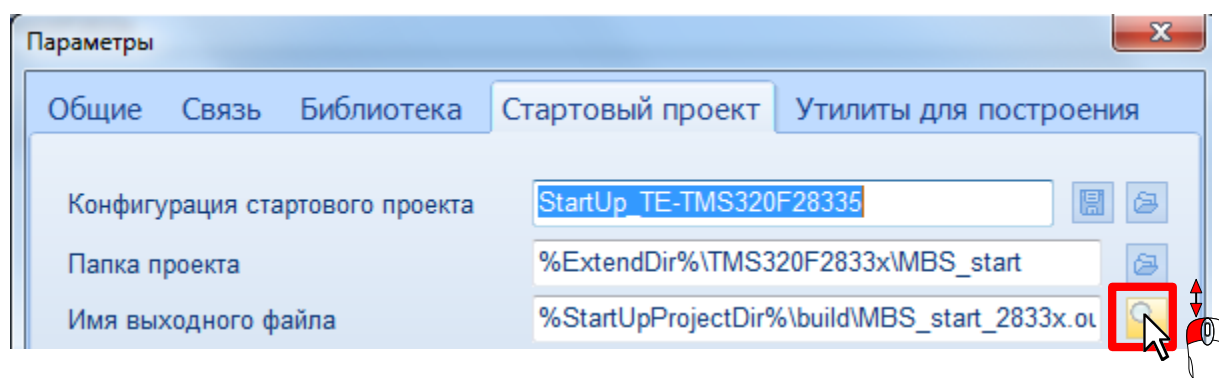
8. В появившемся окне выбрать out файл стартового проекта



Примечание. Для библиотеки процессоров серии **2833x** загружаемый **.out** файл стартового проекта расположен по адресу:

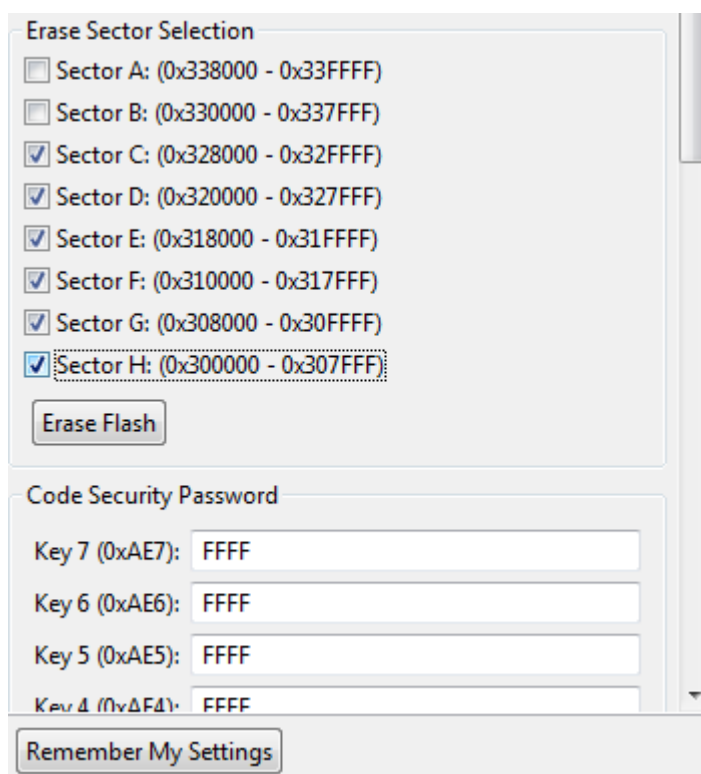
C:\Users\%UserName%\AppData\Roaming\NPF Mechatronica-Pro\MexBIOS Development Studio\Libraries\TMS320F2833x\StartUp\build\MBS_start.out

Папку стартового проекта можно открыть через окно Options в MexBIOS Development Studio:



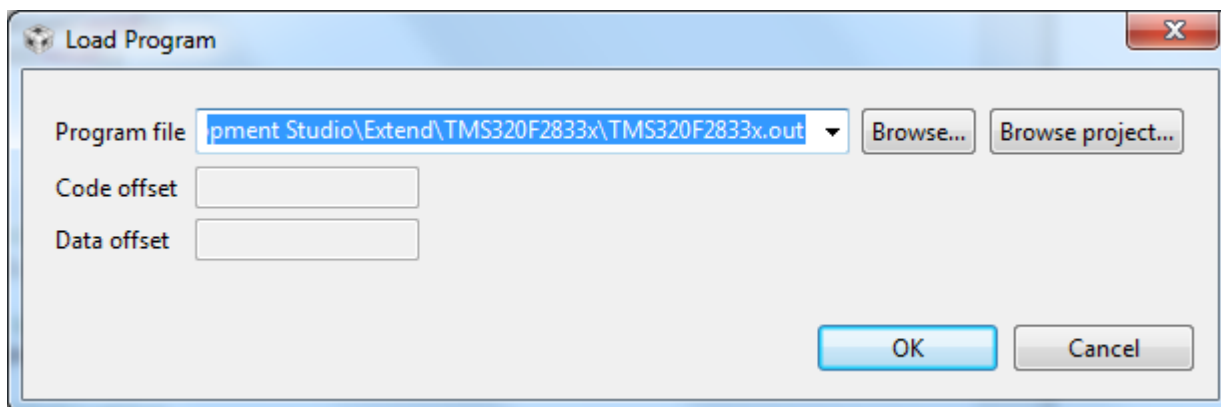
Открытие папки с **Out** файлами стартового проекта для копирования адреса расположения

9. Нажать кнопку **ОК**. Произойдёт загрузка стартового проекта в сектора А и В.
10. Для загрузки библиотеки блоков необходимо изменить настройки **On-Chip Flash**.
 - 10.1. Изменить сектора загрузки, как показано на следующем рисунке. Обязательно нажать кнопку **Remember My Settings**.



11. Нажать кнопку **Load**

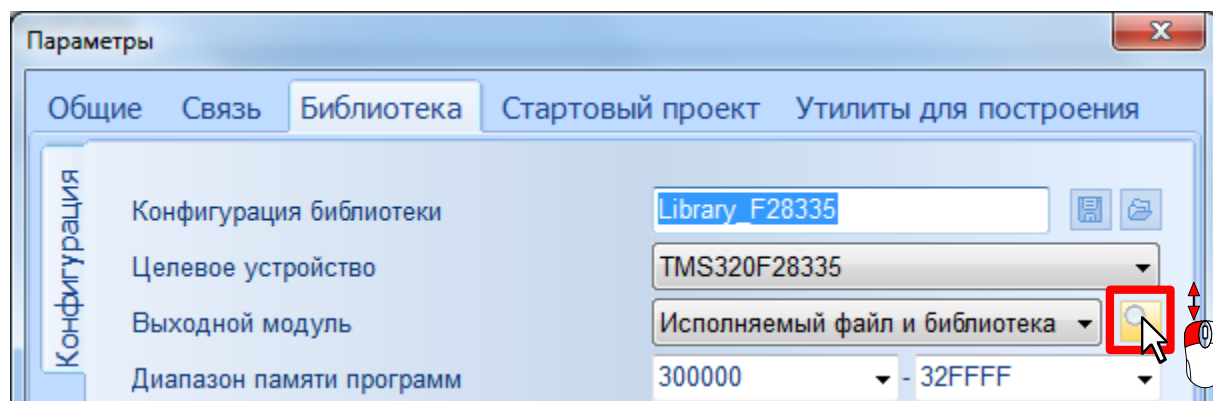
11.1. Выбрать **out** файл библиотеки блоков:



Примечание. Загружаемый **.out** файл библиотеки блоков расположен по адресу:

C:\Users\%UserName%\AppData\Roaming\NPF Mechatronics-Pro\MexBIOS Development Studio\Libraries\TMS320F2833x\TMS320F2833x.out

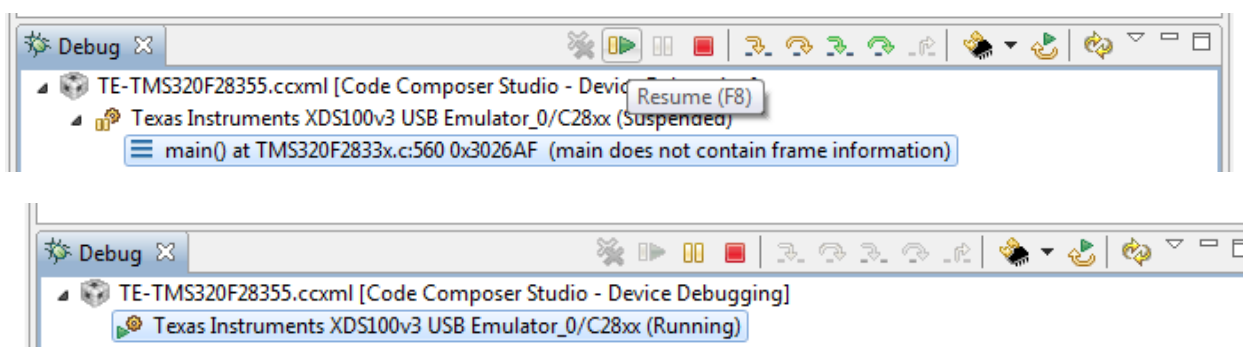
Папку библиотеки проекта можно открыть через окно **Параметры** в MexBIOS Development Studio:



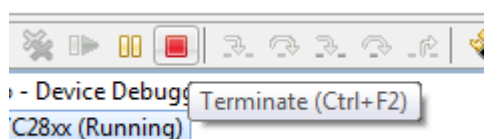
Открытие папки с Out файлами библиотеки для копирования адреса расположения

12. В окне **Load Program** нажать кнопку **OK**. Произойдёт загрузка библиотеки блоков в память контроллера.

13. Далее можно нажать кнопку **Resume** в окне **Debug** и подключиться к плате в окне MexBIOS Development Studio.



14. Также можно нажать **Terminate**. Отключить питание от платы, отключить эмулятор. Закрыть CCS.



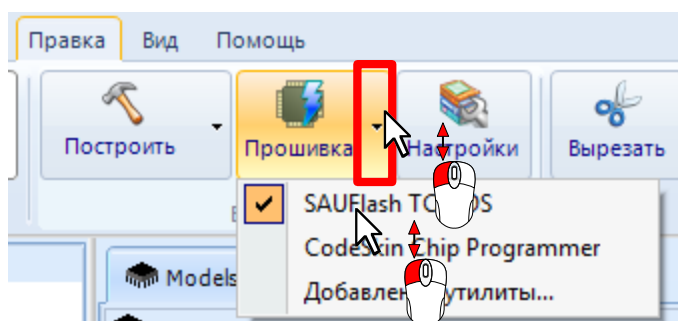
15. Открыть MexBIOS Development Studio. Загрузить один из примеров.

Ядро MexBIOS и библиотека блоков установлены. Аппаратное обеспечение готово для совместной работы с MexBIOS Development Studio.

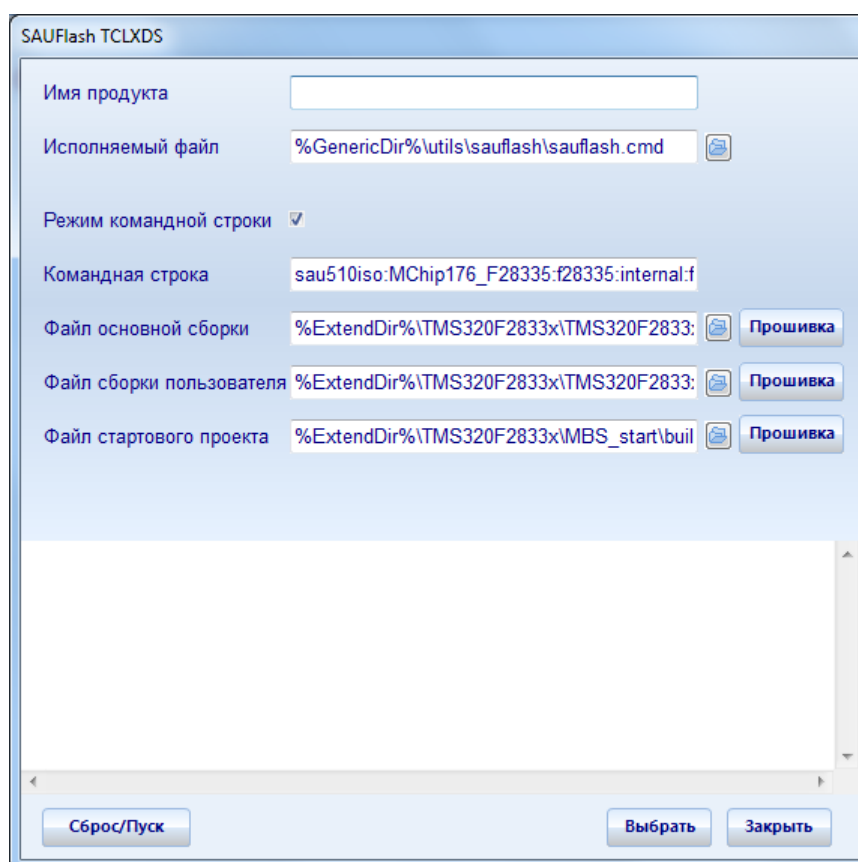
Загрузка стартового проекта и библиотеки блоков с помощью эмулятора SAU510-USB JTAG Emulator

В MexBIOS Development Studio встроен плагин для загрузки скомпилированных файлов во flash память процессора с помощью эмулятора SAU510 USB.

Для использования SAUFlash TCLXDS перейдите на вкладку **Правка**, вызовите дополнительное меню на кнопке **Прошивка**, как показано на следующем рисунке.



Появится диалоговое окно:



1. Подключите эмулятор SAU510 к плате.
2. Нажмите кнопку **Прошивка** пункта **Файл основной сборки**. Произойдёт загрузка



библиотеки блоков во flash память платы.

3. Если не используется библиотека блоков пользователя то **НЕ НУЖНО** нажимать кнопку **Файл сборки пользователя**.
4. Нажмите кнопку **Прошивка** пункта **Файл стартового проекта**. Произойдет загрузка стартового проекта во flash память платы.
5. После загрузки нажать кнопку **Сброс/Пуск** или сбросить питание с платы и отключить эмулятор.

Примечание: Если используется версия эмулятора SAU510 v2, то в пункте **Командная строка** необходимо изменить первый аргумент на Sau510isov2.

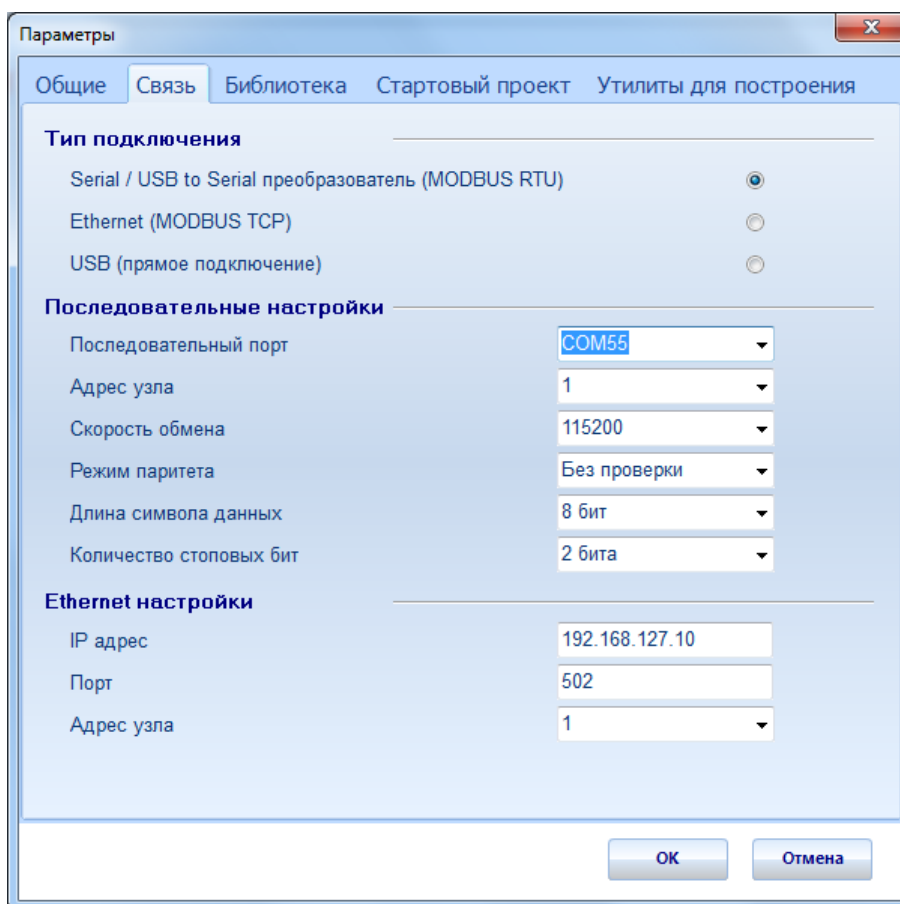
Работа с MexBIOS Development Studio

1. Для загрузки mbs проекта в плату выполните следующие действия:
 - 1.1. Подать питание на плату.
 - 1.2. Подключить USB к плате и компьютеру.
 - 1.3. Запустить **MexBIOS Development Studio**.
 - 1.4. Через стартовое окно нажать кнопку **Открыть пример**. Выбрать файл **2833x_TE-TMS320F28335_TEST.mbp**

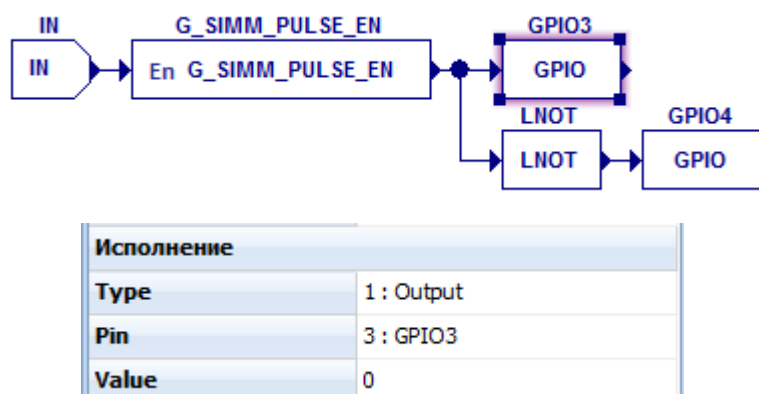


2. Установить связь с платой. Для этого на вкладке **Устройство**, нажать кнопку **Параметры**. В появившемся окне на вкладке **Связь**, в пункте **Последовательный порт** выбрать COM порт, которые создаётся при подключении платы к компьютеру.

Примечание. Если драйвера на плату не установлены, произведите установку драйверов с диска, поставляемого вместе с платой.

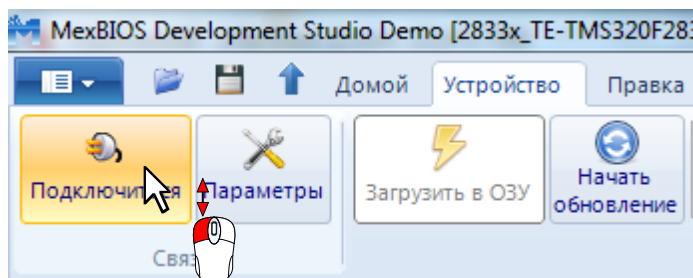


3. В файле проекта изменить на необходимые настройки дискретных ножек GPIO

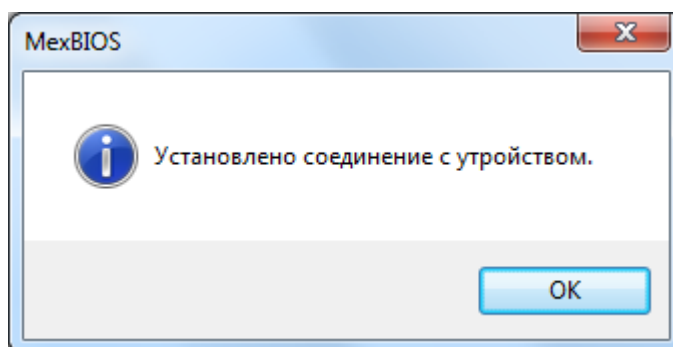


4. Загрузить проект **2833x_TE-TMS320F28335_TEST.mbp**. Пользовательские светодиоды будут мигать на плате.

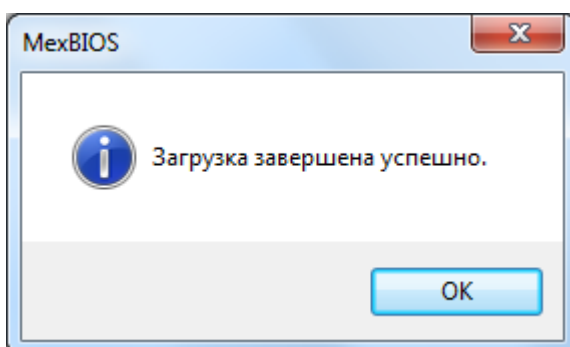
4.1. Для загрузки, перейдите на вкладку **Устройство**.



4.2. Нажмите кнопку **Подключиться**. Нажмите кнопку **Загрузить в ОЗУ**.



4.3. После удачной загрузки появится сообщение:



4.4. Нажмите кнопку **Начать обновление** (либо нажмите горячую клавишу F5) для обмена данными между процессором и программой. В окне **Переменные** можно изменить частоту (только при включенном обновлении) мигания светодиодов в параметре **G_SIMM_PULSE_EN.Time_Pulse**.

Для дальнейшего программирования платы достаточно MexBIOS Development Studio. Если пользователь добавляет свой блок в палитру – необходимо заново скомпилировать библиотеку (вкладка Правка, кнопка Построить) и загрузить библиотеку и стартовый проект средствами CCS либо с помощью плагина SAUFlash TCLXDS.