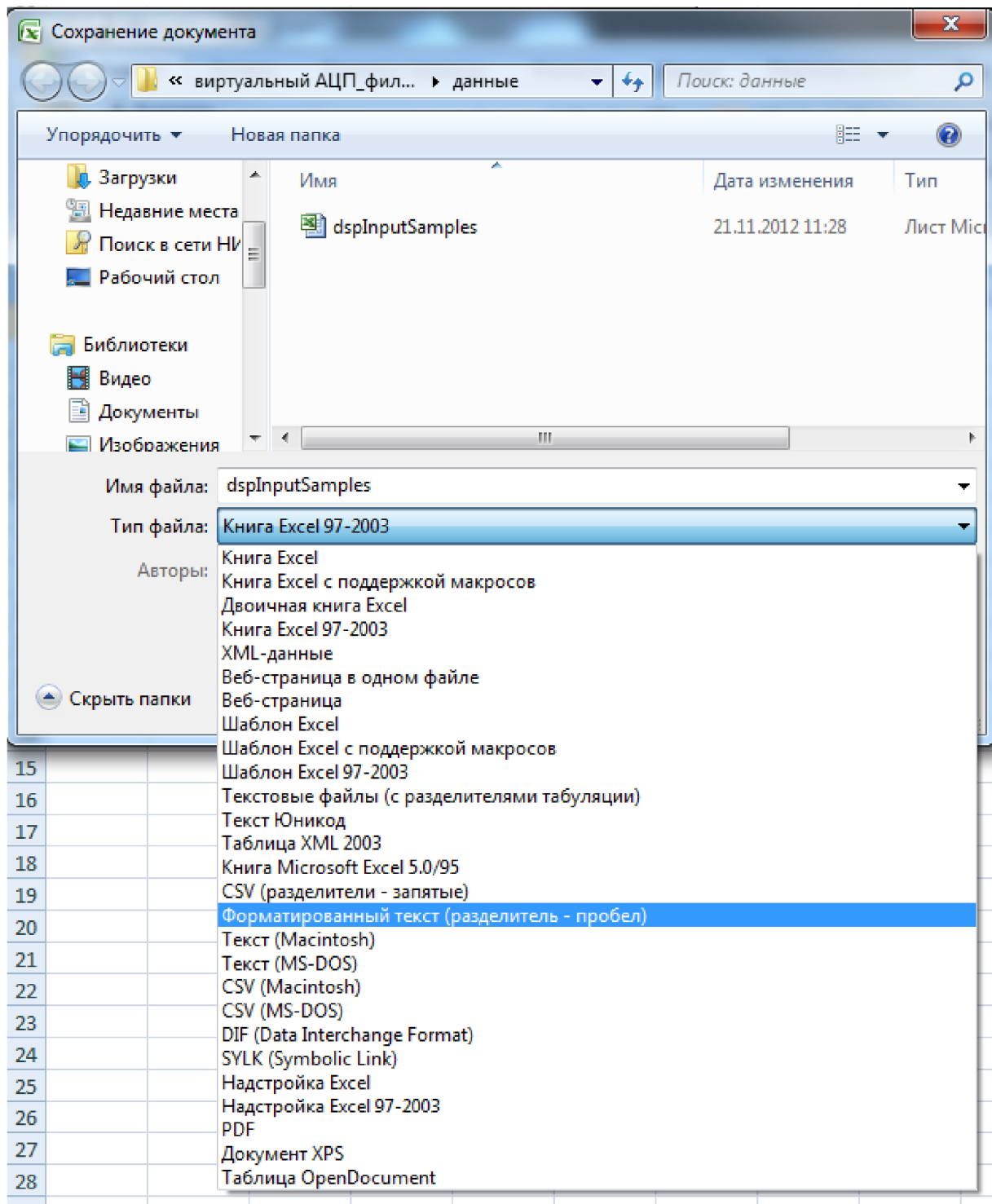
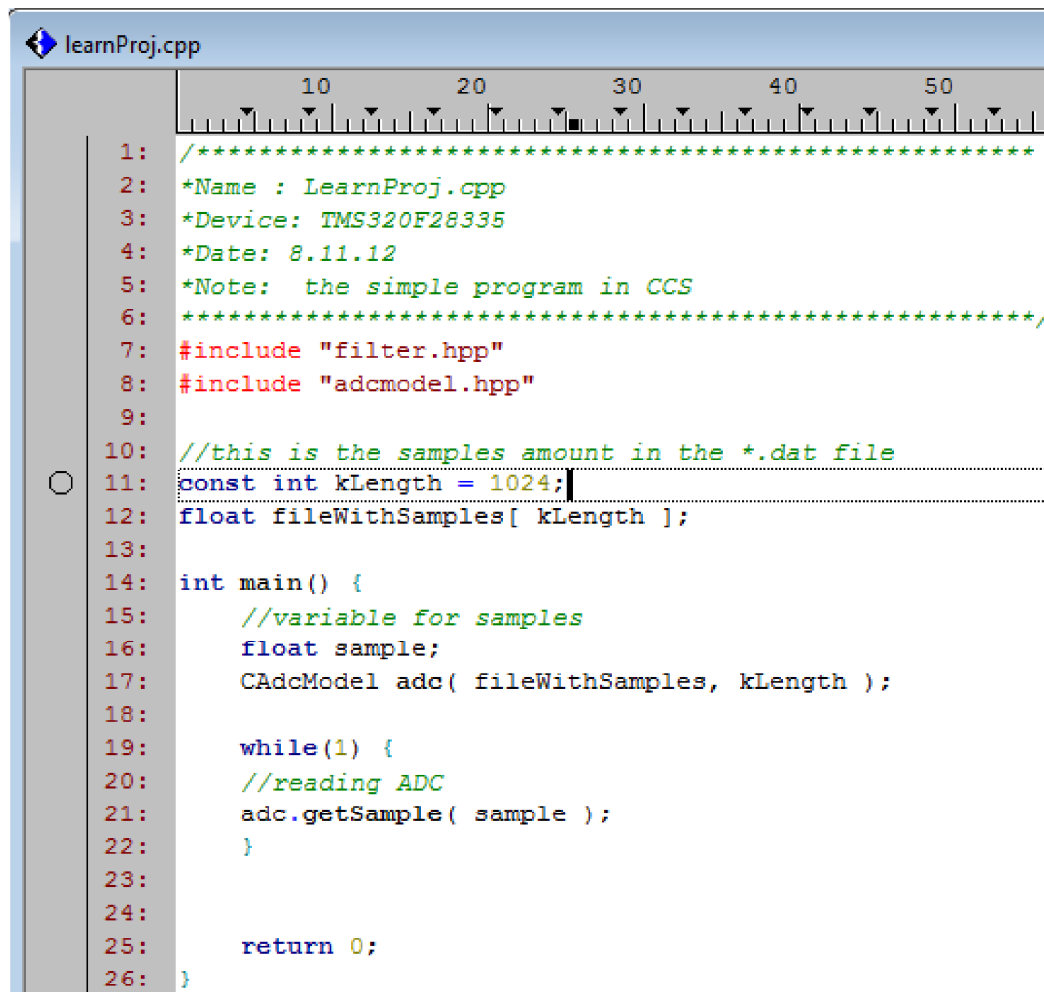




3) *.dat, Code Composer и размещение сэмплов в RAM.

Теперь меняем в программе (см ниже) переменную kLength со значения 1024, на наше значение, т.е. 5387.



```
1:  /*****
2:  *Name : LearnProj.cpp
3:  *Device: TMS320F28335
4:  *Date: 8.11.12
5:  *Note: the simple program in CCS
6:  *****/
7:  #include "filter.hpp"
8:  #include "adcmodel.hpp"
9:
10: //this is the samples amount in the *.dat file
11: const int kLength = 1024;
12: float fileWithSamples[ kLength ];
13:
14: int main() {
15:     //variable for samples
16:     float sample;
17:     CAdcModel adc( fileWithSamples, kLength );
18:
19:     while(1) {
20:         //reading ADC
21:         adc.getSample( sample );
22:     }
23:
24:
25:     return 0;
26: }
```

Проект после изменений не собирается:

```

----- LearnProg.pjt - Debug -----
[learnProj.cpp] "C:\CCStudio_v3.3\C2000\cgtools\bin\cl2000" -g -pdr -pds225 -fr"C:\
"learnProj.cpp", line 19: remark: controlling expression is constant
"learnProj.cpp", line 25: warning: statement is unreachable

[Linking...] "C:\CCStudio_v3.3\C2000\cgtools\bin\cl2000" -@"Debug.lkf"
<Linking>
"C:\tidcs\c28\DSP2833x\v131\DSP2833x_common\cmd\28335_RAM_lnk.cmd", line 131
    run placement fails for object ".ebss"
error: errors encountered during linking; ".\Debug\LearnProg.out" not built

>> Compilation failure

Build Complete,
    2 Errors, 1 Warnings, 1 Remarks.

```

Такая ошибка (*run placement fails for object ".ebss"*
error: errors encountered during linking; ".\Debug\LearnProg.out" not built) обычно
 сигнализирует о невозможности разместить большой кусок кода/данных (а в нашем
 случае именно данных - массива fileWithSamples) в RAM памяти при существующей
 настройке файла линковщика *.cmd. Если быть точным, то не может разместить
 глобальную переменную (массив наш) в блок памяти .ebss, который отвечает за
 глобальные и статические переменные. Подробнее о секциях в памяти смотрим в 7.1.1
 Sections в [SPRU514 TMS320C28x Optimizing C/C++ Compiler](#)

Нам придется править *.cmd Файл, поэтому скопируем его в папку проекта. Для
 вкуривания как править, надо разобраться с Memory Map (Figure 3-2. F28335/F28235
 Memory Map в [SPRS439](#)), а также со структурой cmd-файла (как писать *.cmd Файл
 знакомимся в [SPRU513 TMS320C28x Assembly Language Tools](#) , начиная с 7.5 Linker
 Command Files) .

Наши 5383 сэмпла не помещаются ни в одну именованную 4К секцию памяти в
 RAM. В текущей версии *.cmd наш глобальный массив должен помещаться в:

```
.ebss : > RAML4, PAGE = 1
```

который занимает 4К памяти.

Чтобы справиться с проблемой нужно пробовать способы описанные в 7.5.4.6
 Allocation Using Multiple Memory Ranges и 7.5.4.7 Automatic Splitting of Output Sections
 Among Non-Contiguous Memory Ranges (spru513).

В параграфе 7.5.4.7 сказано, что линковщик может разделять на части выходную
 секцию (например, такую, как наша ".ebss", куда не помещается массив) и помещать их в
 различные области памяти для эффективного размещения данной выходной секции. Для
 создания особого указания линковщику, что данная секция может быть разделена
 используют оператор >> . Поэтому наш файл мы изменим:

```
.ebss : { *(.ebss) } >> RAML4 | RAML5 | RAML6 | RAML7, PAGE = 1
```

Но после сборки проекта выдается та же ошибка. Смотрим внимательней
 документацию, в ней сказано:

“Certain sections should not be split:

- Certain sections created by the compiler, including
 - The .cinit section, which contains the autoinitialization table for C/C++ programs
 - The .pinit section, which contains the list of global constructors for C++ programs
 - **The .bss section, which defines global variables** ”

Поэтому делаем наш массив неглобальным (помещаем в `main(){}`). Файл собрался без ошибок. Строчку в командном файле для секции `ebss` возвращаем в исходное состояние и пересобираем проект. Проект успешно собрался.

Вывод-вопрос: если глобальный массив превысил размеры какой-либо области в памяти то неужели нет никаких средств у линковщика разделить выходную секцию `ebss` (секцию глобальных и статических переменных) и все-таки вместить большую глобальную структуру в RAM ??? или единственный выход помещать большие глобальные структуры данных во FLASH (тем более, что “ Global variables declared far are placed in a section called .ebss. This section can then be linked anywhere in the TMS320C28x data address space” (spru514)??? (После отладки, когда прибор уже выпускается на продажу, все константы, инициализирующие значения и код программы должны храниться во Flash! см. 6.6 Linking Sections to Memory в Module 3: Program Development Tools in the TMS320F28335, Frank Bormann)

Приведу пример того, какие части нашей программы соотносятся с какой секцией памяти (рис. ниже):



C – Compiler Sections

Global vars (`.ebss`)

Init values (`.cinit`)

```
unsigned int k = 0;
unsigned int i;
void main(void)
{
    while(1)
    {
        for(i=0; i<100; i++)
        {
            k=i*i;
        }
    }
}
```

Local variables,
System Context
(`.stack`)

Code (`.text`)

Вопрос: Странно, почему после переноса массива из глобальной области видимости в локальную :

```

11: const int kLength = 5387;
12:
13: int main() {
14:
15:     float fileWithSamples[ kLength ];
16:
17:     //variable for samples
18:     float sample;
19:     CAdcModel adc( fileWithSamples, kLength );
20:
21:     while(1) {
22:         //reading ADC
23:         adc.getSample( sample );
24:     }
25:
26:
27:     return 0;
28: }

```

сборка проекта происходит нормально, хотя 5387 превышает размер выделенный под **.stack** в 28335_RAM_Ink.cmd (стандартный файл TI) ????

MEMORY

{

PAGE 0 :

<..>

PAGE 1 :

<..>

RAMM1 : origin = 0x000400, length = 0x000400 /* on-chip RAM block M1 */

<..>

}

SECTIONS

{

<..>

.stack : > RAMM1, PAGE = 1

<..>

}

Ответ: пока нет.

Но мы можем поэкспериментировать. Ставим размер массива = 100000 => при сборке получаем ошибку: ">> INTERNAL ERROR: Space required for local variables exceeds maximum in _main".

размер массива = 32763 => та же ошибка;

размер массива = 32762 => сборка завершена успешно

32762 = 0x7FFA в 28335_RAM_Ink.cmd нет области такого размера.