

В. Тимофеев

osa@pic24.ru

Как оформлять модули

(памятка для начинающих Си-программистов)

1. Вступление

Для некоторых программистов, привыкших писать текст программы одним файлом (или включать в Си-файл другие Си-файлы директивой `#include`), вызывает трудность оформление и подключение независимых модулей, которые были бы изолированы от основной программы и легко переносились бы в другие проекты. Здесь я опишу, как это делается.

Итак, создается пара файлов с одинаковыми именами и с расширениями `.c` и `.h` (одинаковые имена - необязательное условие, но его нарушение приведет к путанице), например `**new_module.c**` и `**new_module.h**`. Формат и содержание их описан ниже.

2. Содержание основного файла (.c)

```

//*****
// Секция include: здесь подключается заголовочный файл к модулю
//*****

#include "new_module.h"          // Включаем файл заголовка для нашего модуля

//*****
// Секция определения переменных, используемых в модуле
//*****

//-----
// Глобальные
//-----

char GlobalVar1;
char GlobalVar2;
...

//-----
// Локальные
//-----

static char LocalVar1;
static char LocalVar2;
...

//*****
// Секция прототипов локальных функций
//*****
void local_func1 (void);
void local_func2 (void);
...

//*****
// Секция описания функций (сначала глобальных, потом локальных)
//*****
void global_func1 (void)
{
    ...;
}

void global_func1 (void)
{
    ...;
}
...

void local_func1 (void)
{
    ...;
}

void local_func1 (void)
{
    ...;
}
...
//*****
// ENF OF FILE
//*****
```

Описание секций

- **Секция include** - В этой секции подключается заголовочный файл от этого-же модуля. Все остальные заголовочные файлы лучше включать в .h-файле, т.к. они, помимо всего прочего, могут иметь описание типов и констант, которые могут использоваться заголовочным файлом нашего модуля.
- **Секция определения переменных** - Здесь определяются переменные, используемые модулем. Причем для наглядности сперва описываются глобальные переменные (те, область видимости которых будет распространяться на другие модули), а затем - локальные (т.е. те, доступ к которым может осуществляться только внутри данного модуля)
- **Секция прототипов локальных функций** - Здесь описываются прототипы всех локальных функций данного модуля, т.е. тех функций, которые используются внутри самого модуля и вызываются только функциями этого же модуля.
- **Секция описания функций** - В этой секции уже идет оперативная часть кода, т.е. описание самих функций. Причем для удобства желательно также соблюдать последовательность: сначала глобальные, затем локальные.

3. Содержание заголовочного файла (.h)

```
#ifndef _NEW_MODULE_H           // Блокируем повторное включение этого модуля
#define _NEW_MODULE_H

//*****
// Секция include: здесь подключаются заголовочные файлы используемых модулей
//*****
#include <math.h>
#include <stdio.h>
...

//*****
// Секция определения констант
//*****
#define MY_CONST1           1
#define MY_CONST2           2
#define ...

//*****
// Секция определения типов
//*****
typedef struct
{
    ...
} T_STRUCT;

typedef ...

//*****
// Секция определения глобальных переменных
//*****
extern char GlobalVar1;
extern char GlobalVar2;
extern ...

//*****
// Секция прототипов глобальных функций
//*****
void global_func1 (void);
void global_func2 (void);
...

//*****
// Секция определения макросов
//*****
#define MACRO1           ...
#define MACRO2           ...
#define ...

#endif                          // Закрывающий #endif к блокировке повторного включения

//*****
// ENF OF FILE
//*****
```

Описание секций

- **Блокировка повторного включение этого модуля** - один и тот же h-файл может быть включен в несколько модулей, которые также включают заголовочные файлы друг друга. Таким образом получается, что с точки зрения компилятора файл может быть включен в другой файл два или более раз. Тогда бы получалось, что все типы и константы также описаны более одного раза, что вызовет ошибку компилятора (переопределение констант, переопределение типов и т.п.). Чтобы этого не происходило, весь заголовочный файл заключается в скобки `#ifndef...#endif`. Если компилятор видит, что константа `_NEW_MODULE_H` не определена, то он включает весь текст файла, в котором, помимо всего прочего, и определяется константа `_NEW_MODULE_H`. При повторном включении файла компилятор уже видит, что эта константа включена, и все, что заключено в скобки `#ifndef...#endif`, будет им проигнорировано. (*Примечание: имя константы `_NEW_MODULE_H` должно быть свое для каждого модуля. Для исключения путаницы и возможного повторения имен рекомендуется в качестве имени этой константы брать имя файла в верхнем регистре с суффиксом `_H`. Например: в файле `my_module.h` определяем константу `_MY_MODULE_H`; в файле `keyboard.h` определяем константу `_KEYBOARD_H` и т.д.)*)
- **Секция `include`** - здесь подключаются заголовочные файлы модулей, используемых нашим модулем.
- **Секция определения констант** - определяются константы, используемые модулем (или определяющие режим работы модуля). Эти константы будут доступны как самому модулю, так и все модулям, включающим этот файл. Если какую-то константу нужно скрыть (например, она используется только в этом модуле, а есть вероятность, что в каком-нибудь стороннем модуле встретится константа с таким же именем), то ее определение можно перенести в основной файл.
- **Секция определения типов** - здесь определяются все специфичные для этого модуля типы данных.
- **Секция определения глобальных переменных** - здесь описываются переменные, которые будут доступны из других модулей. Следует обратить внимание на то, что в заголовочном файле переменные описываются с обязательным квалификатором "extern".
- **Секция прототипов глобальных функций** - Здесь описываются прототипы функций, которые будут видны другим модулям, чтобы компилятор знал, с какими параметрами их можно вызывать.
- **Секция определения макросов** - здесь можно описать какие-то присущие модулю макроопределения.

4. Подключение к проекту

Чтобы включить созданные файлы в свой проект нужно:

1. В интегрированной среде добавить в проект оба файла (`new_module.c` и `new_module.h`). Можно ограничиться только си-файлом, но для удобства работы с рабочей областью (`workspace`) лучше добавлять оба. (Например, в MPLAB добавление файлов делается через меню: Project/Add Files to Project).
2. Во всех файлах проекта, в которых предполагается использование функций, типов, переменных, констант или макросов из новых файлов, вставить строчку:

```
#include "new_module.h"
```

5. Примечания

Примечание 1

Все секции, кроме `"include"` в си-файле и "блокировки повторного включения" в h-файле, являются необязательными, однако, даже если какой-то секции нет (например, в модуле нет локальных функций), то лучше комментарий, описывающий секцию оставить, чтобы при просмотре файла не возникало вопросов: а где эти локальные функции могут быть описаны? а вдруг они где-то в другом месте? и т.п. А так сразу видно, что есть секция, но она пустая, следовательно, локальных функций нет.

Примечание 2

Порядок секций, последовательность описаний внутри секций не имеет значения, если, конечно, не нарушается порядок, определенный стандартом Си (например, прототип функции должен быть описан в файле раньше, чем первое к ней обращение). Кроме того, если описания будут перемешаны между собой (например, сначала идут переменные, потом прототипы, потом опять переменные и т.п.), то ничего страшного, кроме потери наглядности, не произойдет.

Примечание 3

При необходимости можно добавлять другие секции (например, иногда нужно описать несколько констант внутри основного си-файла).

Примечание 4

Обратим внимание, что в си-файле нет секции описания прототипов глобальных функций. Их туда можно было бы добавить, но получится, что они просто будут дублировать уже описанные прототипы в h-файле (тут они должны быть обязательно, иначе остальные модули не будут знать про эти функции).

Это несколько неудобно, т.к. при смене спецификации функций исправления в прототипах придется делать в двух местах.

Виктор Тимофеев, июнь, 2010

osa@pic24.ru