Проектиране на вградени системи
Лабораторно упражнение №1

Работа с Makefile и Menuconfig. Йерархични Makefile-ове.
Документиране на сорс код с Doxygen.

==========

1. Да се разучи вътрешната структура на микроконтролера STM32F769I (базиран на ARM Cortex-M7).

2. Да се разучи принципната схема на демо платката STM32F769I-DISCO.

3. Компилирайте и линкнете програма за мигане на светодиода LD3, дадена в директория **01_03**. За целта отворете терминал с CTRL + T, влезте в работната ви директория и изпълнете командите:

```
arm-none-eabi-gcc -mcpu=cortex-m7 -mthumb -nostartfiles
--specs=nosys.specs -Tscript.ld main.c -o main.axf

arm-none-eabi-objcopy -O binary main.axf main.bin
```

4. Заредете програмата main.bin във флаш паметта на микроконтролера STM32F769I посредством сървърното приложение OpenOCD. За целта изпълнете командата:

```
openocd -f board/stm32f7discovery.cfg -c"init" -c"reset halt"
-c"flash probe 0" -c"flash write_image erase main.bin 0x08000000"
-c"reset run" -c"exit"
```

Ако програмирането е успешно светодиод LD3 трябва да започне да мига.

5. Напишете Makefile, който да автоматизира процеса на компилация, линкване и зареждане. Нека обектовите и двоичните файлове да се преместят в отделна директория с име “debug”. Имплементирайте следните Makefile target-и:

```
make
make flash
make clean
```

6. Разделете кода на две части – main.c с основния алгоритъм, led.c с функция за инициализация и включване/изключване на светодиода, и startup.c с векторната таблица и кода на ресет хендлера. Коригирайте Makefile-а, за да отразите промените.

7. Преместете файловете на светодиода в отделна директория с име “led”.

Създайте отделна директория с име “uart” и копирайте файловете, инициализиращи един от UART модулите на STM32F769I, дадени в директория **01_07**. От същата директория копирайте сорс файловете и създайте отделна директория с име “printf”, имплементираща минималистична printf функция. Нека всяка директория си има Makefile. Създайте един основен Makefile в top-level директорията, който да вика отделните Makefile-ове [1]. Препоръчително е да използвате променливата ${MAKE} вместо командата make за рекурсивните target-и [2].

За да спрете GCC на ниво компилатор (без да линквате), използвайте аргументът -c. Изходният файл тогава трябва да бъде обектов файл с разширение “.o”, като например:

```
arm-none-eabi-gcc -mcpu=cortex-m7 -mthumb -nostartfiles
--specs=nosys.specs -c uart.c -o uart.o
```

Добавете пътищата до отделните C файлове посредством аргумента -I, когато компилирате main.c:

```
arm-none-eabi-gcc -mcpu=cortex-m7 -mthumb -nostartfiles
--specs=nosys.specs -I./uart -I./led -I./printf -c main.c -o
./debug/main.o
```

8. Направете примерът ви да може да се конфигурира от графичната програма (от командния ред) – lxdialog (mconf), която използва ncurses библиотеката. Тя създава конфигурационен файл .config, който се преобразува във хедърен файл от програмата conf. Този хедър ще се използва, за да конфигурира програмата ви. За целта първо трябва да компилирате mconf и conf. Копирайте директорията **menuconfig** [3] от директорията **01_08** в top-level директорията на проекта ви. След това изпълнете командите:

```
cd menuconfig/
mkdir build
cd ./build
cmake ../
make
```

(ако инсталацията ви е нова може да се наложи да инсталирате ncurses: sudo apt-get install libncurses-dev)

Ако всичко е минало без грешки изпробвайте програмата като изпълните командата:

```
./mconf ../test/rootconf
```

Преобразуването на автоматично генерираният файл menuconfig/build/.config в хедърен файл става посредством командата (прието е този хедър да се нарича

autoconf.h) [4]:

```
./conf ../../Kconfig --silentoldconfig autoconf.h
```

9. Разгледайте структурата на файла menuconfig/test/rootconf. Опитайте се да създадете аналогичен файл за вашия проект с име Kconfig, който да конфигурира следните модули:

*__led__ – възможност за контрол на паузите между премигванията (50000, 200000, 400000 итерации на for-цикъла);
*__uart__ – възможност за контрол на скоростта на предаваните данни (9600, 19200, 115200 kbit/s)

За целта направете софтуерното закъснение на паузите на светодиода, както и инициализацията на UART-а с макроси. Имената на тези макроси трябва да съответстват на имената в генерирания autoconf.h. Не забравяйте да включите:

```
#include “autoconf.h”
```

във вашата програма.

ВНИМАНИЕ: понеже menuconfig target-а е команда, която не зависи от друг файл и трябва да се изпълнява винаги, направете я “PHONY”:

```
.PHONY: menuconfig
```

Без този ред в Makefile-а винаги ще получавате следното съобщение:

```
user@computer:~/01_09$ make menuconfig
make: `menuconfig' is up to date.
```

10. Документирайте кода с Doxygen коментари (програмата Doxygen може да се инсталира в Ubuntu Linux със: sudo apt-get install doxygen) [6]. Основната структура на един такъв коментар изглежда така:

```
 /*!
  * \brief Това е кратко описание на функцията.
  *
  * През един ред следва детайлно описание на функцията
  * заедно с всичките ѝ особености и тънкости. Ако тази
  * функция засяга други, те могат да бъдат изредени тук.
  * Ако тази функция засяга глобални променливи, добре е
  * да се опише това. Може да се използват и HTML тагове.
  * Например \bтази дума ще е засветена.
  *
  * \param my_value – кратко описание на параметъра
  *
  * \return 1 – успех, 0 - неуспех.
  */
int my_function(char my_value){
```

```
…
}

/*!
  * \struct my_structure
  * \brief Това е кратко описание на променлива от тип
  * структурата
  *
  * През един ред следва детайлно описание.
  *
  */
struct my_structure;
```

Други типове, файлове, функции и полета също могат да се документират: \struct, \union , \enum, \fn, \var, \def, \typedef, \file, \namespace.

След като документирате всички променливи, функции и макроси, отворете команден ред, преместете се в top-level директорията на проекта ви и изпълнете:

```
doxygen -g
doxygen Doxyfile
cd ./latex
make
```

Документацията ще се генерира в директорията latex и ще е с име refman.pdf. На този етап това ще е един празен файл.

11. Използвайте графичната програма Doxywizard (може да се инсталира в Ubuntu Linux със: sudo apt-get install doxygen-gui), за да конфигурирате форматирането на вашия PDF файл. За целта влезте в top-level директорията ви и напишете:

doxywizard Doxyfile

Следвайте менютата на doxywizard. Добавете логото на ТУ-София от директория **01_11**. Като изходна директория за документацията изберете latex. Когато сте готови, на таб-а Run натиснете Run doxygen. Затворете графичното приложение, влезте в съответната директория и стартирайте Make:

cd latex/latex/
make

* * *

[1] https://www.gnu.org/software/make/manual/html_node/Recursion.html
[2] https://www.gnu.org/software/make/manual/html_node/MAKE-Variable.html#MAKE-Variable
[3] https://github.com/TheGeorge/menuconfig
[4] https://johnvidler.co.uk/linux-journal/LJ/222/11333.html

[5] https://www.gnu.org/software/make/manual/html_node/Phony-Targets.html
[6] https://www.doxygen.nl/manual/commands.html#cmdcond

гл. ас. д-р инж. Любомир Богданов, 2021 г.