

Факултет по Математика и Информатика
Катедра Софтуерни технологии
Дипломна Работа на тема:
“Статична C библиотека за Arduino”
Дипломант:
Веселин Станчев
ФН: 1801321012
спец. СИ

Увод

С развитието на IoT все повече стават популярни едноплатковите development boards като Arduino Uno и Raspberry Pi Pico, които могат да послужат за учебни, университетски проекти или домашна автоматизация. Съществуват няколко вида instruction sets:

- CISC -> Complex Instruction Set Computer
- RISC -> Redused Instruction Set Computer
- CISC се използва при x86_64 базираните настолни компютри и лаптопи.
- RISC се използва при микроконтролерите Atmega 328p и RP2040.

Начини за програмирането на Arduino

Начините за програмирането на Arduino са:

- чрез използването на C++ базирания диалект
- чрез използването на C
- чрез използването на Assembler

Езикът C е език от ниско ниво и затова гарантира бързина на изпълнение на програмата. Много по-добре е да се използва C, отколкото официалния C++ диалект.

Ако се цели още по-голяма бързина, тогава се използва Assembler.

Цел на дипломната работа

- Да бъде създадена статична библиотека на C за Ардуино заедно със Асемблерска част, която да бъде включена в библиотеката. Целта на библиотеката е да покаже взаимодействието между C и Assembler.

Причини за изпълнение на поставената цел

- Търсене на по-бърз и ефективен начин за програмиране на микроконтролерите
- Алтернативен начин за създаване на статична библиотека – с основна асемблерска част
- Използване на софтуерни инструменти независими от конкретен език

Произтичащи от целта задачи

- Изследване на съществуващите C библиотеки и анализ на datasheet-овете на микроконтролерите, за които е предназначена асемблерската част
- Анализ на целевите процесорни архитектури, за които е предназначена библиотеката.
- Дефиниране на изискванията към библиотеката
- Използвани софтуерни инструменти
- Кодиране на библиотеката
- Постигнати резултати. Бъдещо Развитие.

1. Изследване на съществуващите C библиотеки и анализ на datasheet-овете на микроконтролерите, за които е предназначена асемблерската част

Съществуват 2 типа библиотечни файлове (библиотеки) – статични и динамични. Статичната библиотека представлява архив с разширение .a , който се състои от обектни файлове с разширение .o .

Динамичната библиотека от своя страна представлява файл с разширение .so (shared object) .

Примери за С библиотеки

- `stdlib.h`
- `stdio`
- `math.h`

Кратък анализ на datasheet-а на Atmega 328P

- Според datasheet-а Atmega 328P е 8-bit RISC базиран микроконтролер. Може да бъде използван GNU Assembler-а, който е съвместим с RISC-базирани устройства.

2. Анализ на ARM архитектурата

- ARM архитектурата разполага с общо 32 регистъра. Има регистри с общо и специално предназначение.
- Примери за регистри с общо предназначение са:
 - r0
 - r1
 - r2
- Примери за регистри със специално предназначение са:
 - r13 -> Stack pointer
 - r14 -> Link Register
 - r15 -> Program Counter

3. Дефиниране на изискванията към библиотеката

Статичната библиотека трябва да поддържа микроконтролерите:

- Atmega 328P
- RP2040
- За основната асемблерска част на библиотеката е необходимо да се използва
- асемблер, който е съвместим с ARM RISC процесорната архитектура.
- Библиотеката трябва да се състои от следните функции:
функция за обработка на масив
функция за обработка на указател

4. Използвани софтуерни инструменти



За реализацията на проекта е избран сетът от инструменти GNU Tools.

Причината за избора е, че всички инструменти са или с отворен лиценз като редактора Vim или са част от GNU Project.

- Gcc => компилатор, който може да се използва за C, C++,Assembler.
- Make => build система за автоматизация. Използва се за автоматизиране на компилацията за C, C++ и Assembler програми.
- Gdb => дебъгер за C и Assembler. Използва се след като програмата е компилирана с дебъг символи чрез параметъра -g.
- Vim => терминален текстов редактор, който може да се използва за всички програмни езици.
- Git => система за контрол на версиите

5. Кодиране на библиотеката

Функциите за библиотеката ще бъдат реализирани чрез GNU Assembler.

Част от изходния код на асемблерския файл `arrayfunc.s`

- `.data // секция за данните`
`.balign 4 // заделяне на 4 байта в паметта`
`rtrn1: .word 0 // променлива с етикет rtrn1 със стойност 0`
`.balign 4`
`// заделяне на 4 байта в паметта`

Скриншоти на кода

```
1 .data
2
3 .balign 4
4 rtrn1: .word 0
5 .balign 4
6 rtrn2: .word 0
7 array1: .skip 40
8
9 .text
10
11 arrayfunc:
12 ldr r1,=rtrn2
13 str lr,[r1]
14 ldr r2=array1
15 mov r3, #5
16 mov r4,#6
17 ldr lr,=rtrn2
18 ldr lr, [lr]
19 bx lr
```

```
1 .data
2
3 .balign 4
4 rtrn1: .word 0
5 .balign 4
6 rtrn2: .word 0
7 array1: .skip 40
8
9 .text
10 arrayfunc:
11 ldr r1,=rtrn2
12 str lr,[r1]
13 ldr r2=array1
14 mov r3, #5
15 mov r4,#6
16 ldr lr,=rtrn2
17 ldr lr, [lr]
18 bx lr
```

Постигнати Резултати Бъдещо развитие

Участия с темата на дипломната

Темата на дипломната работа стана обект на участие в 2 технически конференции

- Клуб Роботика - ТУ Пловдив
- PlovdivConf 2022

За използването на Асемблер в профилираните гимназии

- Независимо от програмирането насочено към крайния потребител, интересът на учениците към системното програмиране на вградени системи става все по-голям.
- GNU Assembler-a е изключително подходящ за първи стъпки към изучаването на Асемблерен език от ученици и студенти.

Възможности за бъдещо развитие

- Адаптиране на библиотеката към AVR Assembler - за използване на голям брой Atmel-базирани микроконтролери
- Адаптиране на библиотеката към RISC-V архитектурата – за RISC-V базирани микроконтролери

Заклучение

- Изискванията към библиотеката бяха изпълнени.
- Поставените цели бяха успешно изпълнени.
- Целта на дипломната работа е постигната.
- Дипломната работа ще бъде допълнително развита, чрез използване на някои от посочените възможности.

ИЗПОЛЗВАНИ ИЗТОЧНИЦИ

- Arduino Atmega 328P Datasheet
- RP2040 Datasheet
- ARM Architecture Reference Manual
- Linux Man Pages
- Free Software Foundation Manuals
- GNU Assembler Book