Структура HEX-файла: <https://hubstub.ru/stm32/160-pishem-svoy-butloader-dlya-stm32-prodolzhenie.html>

Задачи

:llaaaatt[dd...]cc

Группы состоят из одинаковых букв, а каждая буква соответствует одному шестнадцатеричному числу.

**:** начало записи

**ll** количество данных dd в записи

**aaaa** адрес начала записи

**tt** тип записи:

1. Разобрать собственный хекс, узнать с какие адреса занимает код
2. Установка дополнительного адреса

:020000040800**F2**

: - начало записи

02 – кол-во данных в dd = 2 байта

0000 адрес начала записи

04 – тип записи - содержит дополнительный адрес

|  |
| --- |
| **00** — cодержит данные **01** — последняя запись в файле **~~02~~**~~— дополнительный адрес сегмента~~ **04** — запись содержит дополнительный адрес, необходимы для вычисления полного адреса **~~05~~**~~— адрес начала приложения, а именно функции~~**~~main~~**~~(только для~~**~~MDK-ARM~~**~~)~~ |

0800 – данные (собственно доп адрес)

F2 – чек сумма

01h + NOT(02h + 00h + 00h +04h +08h + 00h) = F2

- **02** количество байт в записи  
- **0000** адресное поле, для данного типа записи всегда равно 0000  
- **04** тип записи  
- **0800** старшие 16 бит адреса  
- **F2** чек сумма записи рассчитывается как  
01h + NOT(02h + 00h + 00h + 04h + 08h + 00h).

Б) завершение программы

:00000001**FF**

В) участок кода

LPC2378

:1000100034F09FE534F09FE530F09FE52CF09FE5**5C**

10h = 16 байт данных

0010 адрес начала записи

00 – содержит данные

34F09FE534F09FE530F09FE52CF09FE5 данные

**5C** – контрольная сумма

1. У меня 32-разрядная система, поэтому с вероятностью 100% устанавливается дополнительный адрес перед каждой секцией

«Скелет» бутлоадера: <https://hubstub.ru/stm32/159-pishim-svoy-butlouder-dlya-stm32.html>