# Организация автоматизированного тестирования встраиваемого программного обеспечения

Кижнеров Павел Александрович
группа 21.M07-мм
руководитель Терехов А. Н.

СПбГУ

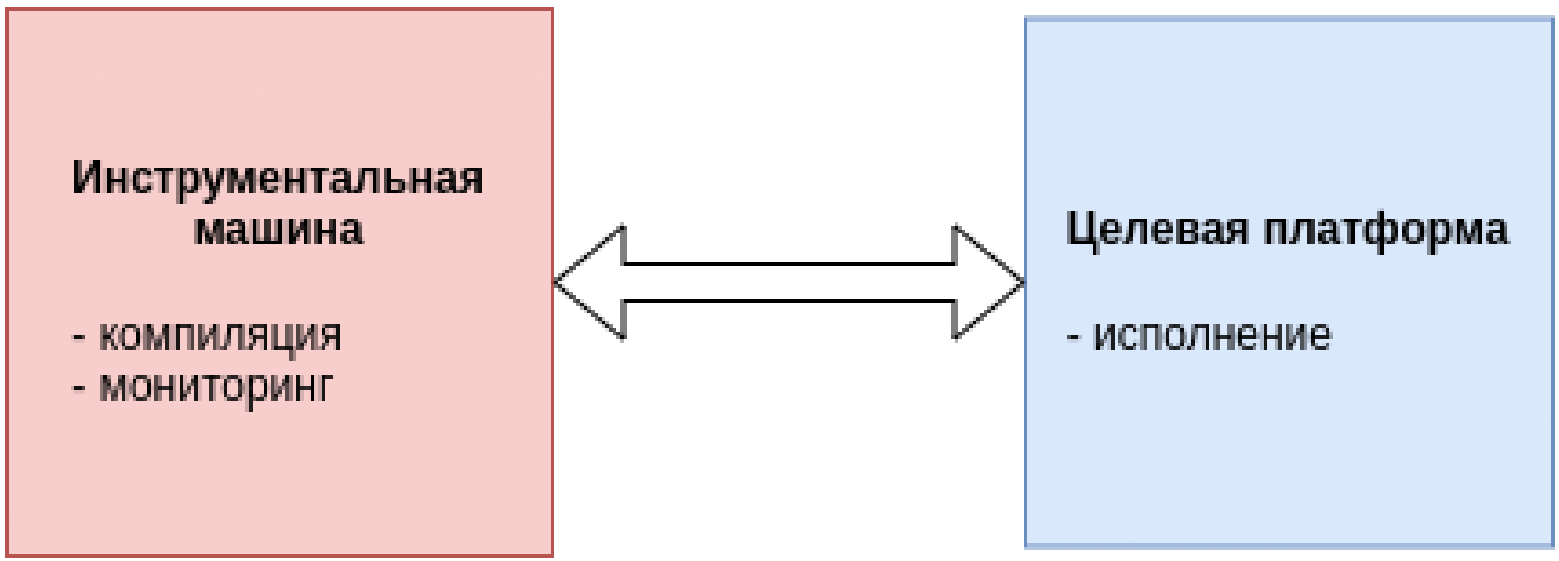
29 декабря 2022 г.

# Введение

Данная работа выполняется в рамках разработки прошивки для фитнес-браслетов

# Введение

Тестирование - неотъемлемая часть контроля качества. Разработка встраиваемого ПО предполагает использование инструментальной машины.



# Постановка задачи

Организовать автоматическое и непрерывное тестирования прошивки с возможностью мониторинга результатов во времени.

# Постановка задачи

- как тестировать платформно-независимый код?
- как тестировать платформно-зависимый код?
- как автоматизировать тестирование и отчетность?
- можно ли автоматизировать генерацию тестов?

# Постановка задачи

Подзадачи:

- организовать тестирование платформно-независимого кода
- организовать тестирование платформно-зависимого кода
  - реализовать интерфейс взаимодействия с целевой платформой
- автоматизировать тестирование и отчетность
- автоматизировать генерацию тестов

# Обзор

Данные и ограничения:

- язык разработки - C
- система сборки - CMake
- инфраструктура - GitLab, Jenkins
- много разработчиков и мало оборудования

# Обзор: автоматизация тестирования и отчетности

Решения определены уже используемыми в компании технологиями - связка GitLab + Jenkins.

Из этого следует ограничение на выбор тестирующих платформ - необходима возможность формировать XML отчет.

# Обзор: тестирование платформно-независимого кода

Классические решения:

- **Gtest**
- Catch
- Mettle
- Boost.Test

# Обзор: тестирование платформно-зависимого кода

Требуется интерфейс взаимодействия с целевой платформой.
Решения:

- **Robot Framework**
- TETware RT
- OpenTest
- autotestnet
- DejaGnu

# Обзор: автоматизация генерации тестов

Технологии:

- **Символьное исполнение (KLEE)**
- Fuzzing

KLEE - стандарт де-факто.

# Реализация: тестирование платформно-независимого кода

Продукт сборки - CMake скрипт, включающий компиляцию указанных платформно-зависимых модулей.

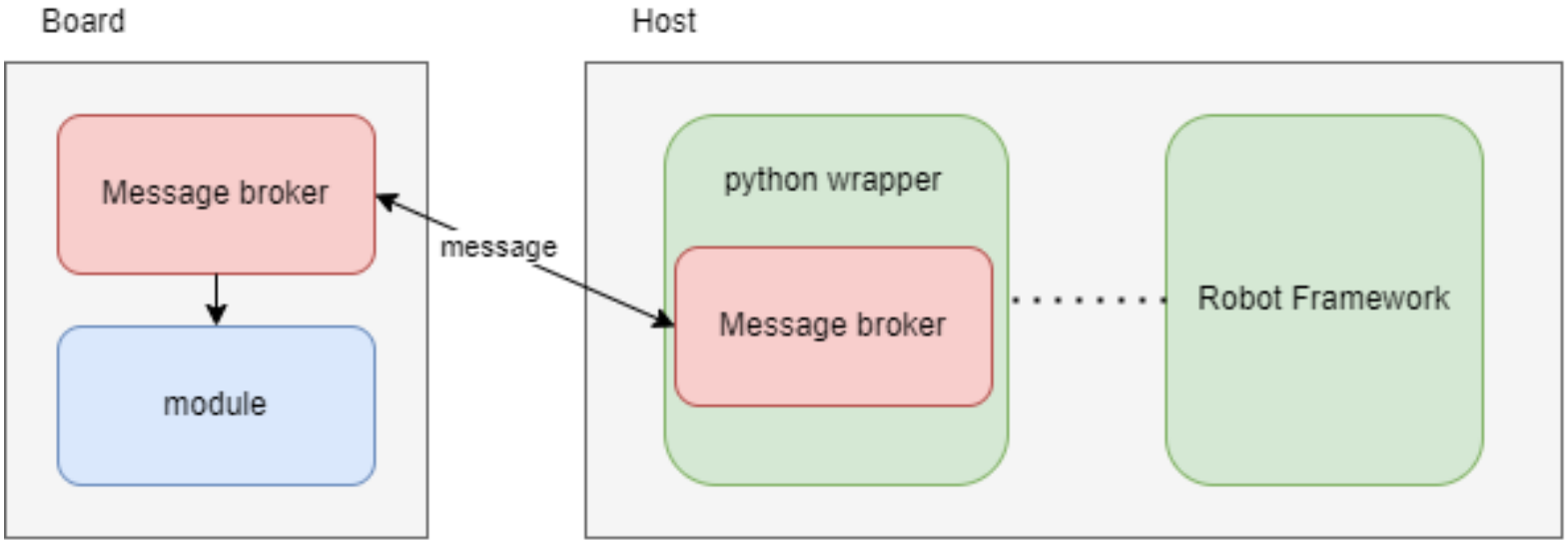
Продукт сборки, не включающий ни одного платформно-зависимого модуля - платформно-независимый.

Может быть собран и исполнен на инструментальной машине.

Интеграция с GTest является естественной.

# Реализация: тестирование платформно-зависимого кода

Готового интерфейса взаимодействия с целефой платформой не удалось найти.
Реализован самостоятельно.




Интеграция с Robot Framework является естественной.

# Реализация: автоматизация генерации тестов

Система сборки была подготовлена для генерации LLVM IR биткода, который необходим для работы KLEE.

Подготовлен модуль, инициирующий символьное исполнение для указанной функциональности.

Работает с платформно-независимым кодом.

# Реализация: автоматизация тестирования и отчетности

Jenkins выполняет сценарии проверки прошивки после каждого изменения в главной ветке репозитория GitLab. Сценарий состоит из:

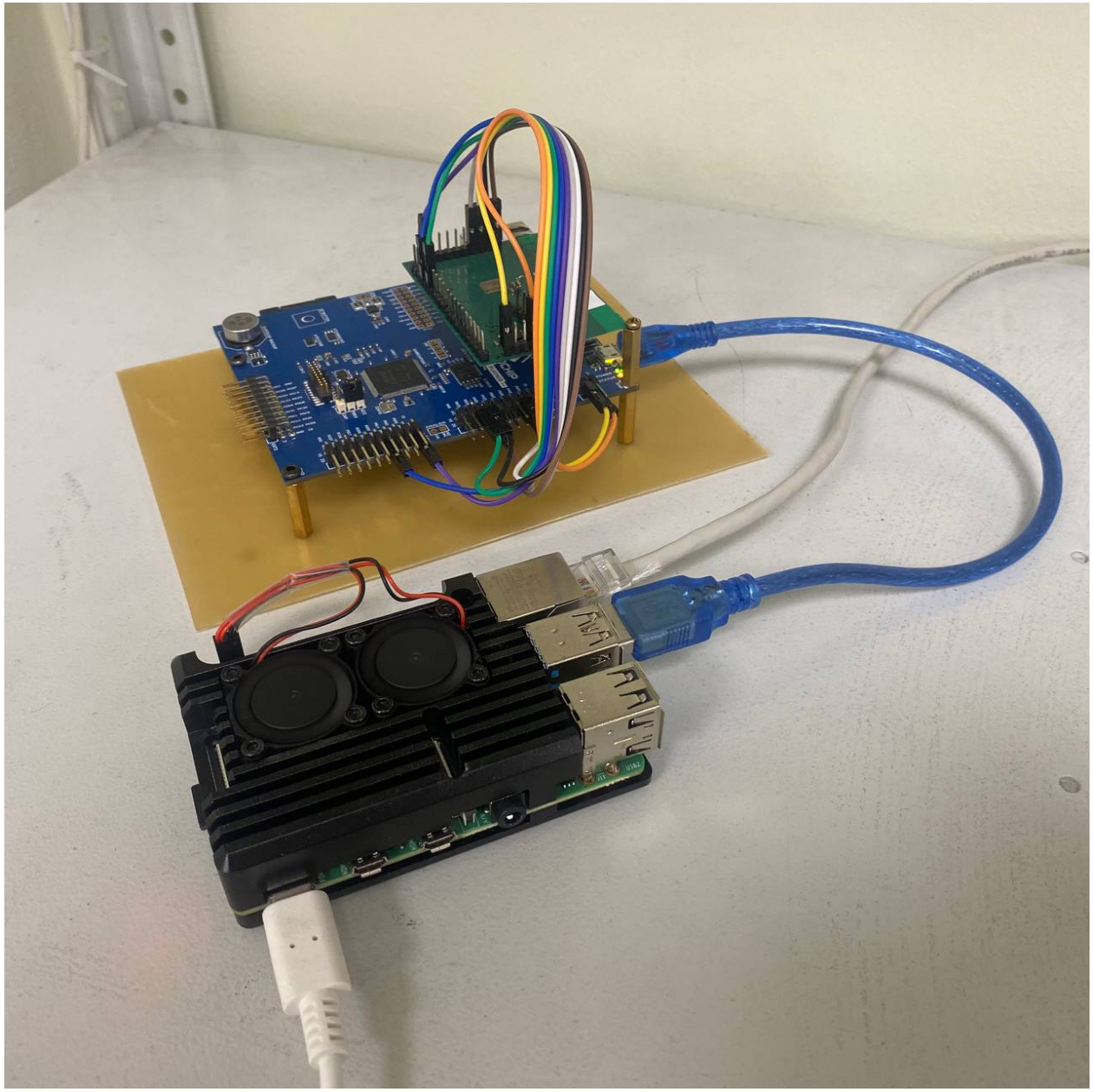
- проверки компилируемости различных продуктов
- платформно-независимого тестирования
- платформно-зависимого тестирования
  - сборки целевого продукта
  - прошивки целевой платы
  - запуска функций прошивки согласно скриптам

# Результаты

- организовано тестирование платформно-независимого кода
- организовано тестирование платформно-зависимого кода
  - реализован интерфейс взаимодействия с целевой платформой
- автоматизированы тестирование и отчетность
- автоматизирована генерация тестов

Система активно используется 8 месяцев.

# Результаты: тестовый стенд



# Результаты: автоматизация тестирования и отчетности

**Pipeline master**

Full project name: gb31_core/master

Last Successful Artifacts

| | | |
|---|---|---|
| documentation.zip | 8.46 MB | view |
| emu-debug.zip | 2.52 MB | view |
| emu-release.zip | 2.15 MB | view |
| xpro-debug-test-on-start.zip | 2.13 MB | view |
| xpro-debug-test.zip | 2.13 MB | view |
| xpro-debug.zip | 1.95 MB | view |
| xpro_bt.zip | 2.12 MB | view |

Recent Changes

Test Result Trend

Failed Skipped Passed

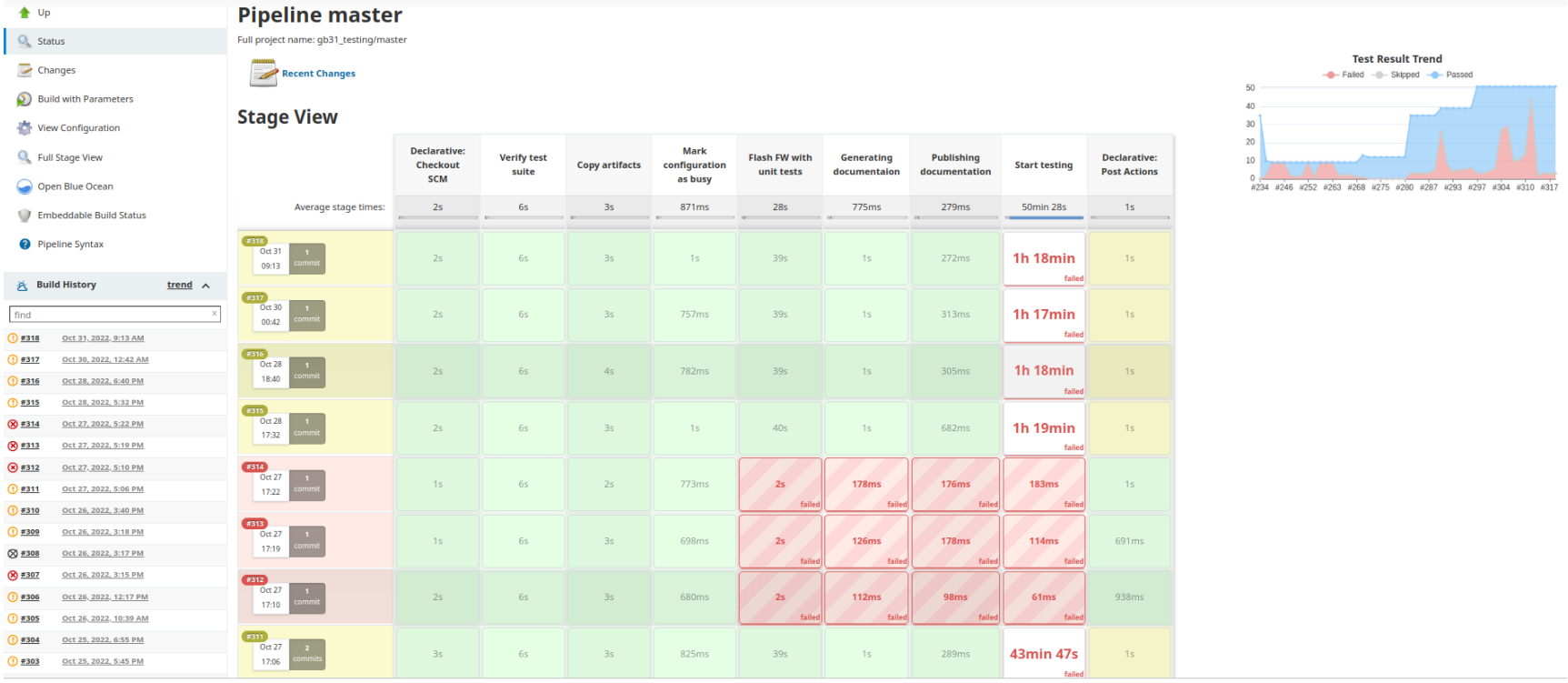
48 40 30 20 10 0

#153 #158 #162 #166 #170 #174 #178 #182 #186 #190 #194 #199 #203

**Stage View**

| | Declarative: Checkout SCM | Declarative: Agent Setup | Prepare build folder | Build for all boards | xpro (Debug) | xpro (Debug Test On Start) | xpro (Debug Test) | xpro_bt (Debug, BT, Test) | emu (Debug) | emu (Release) | emu(Debug Test) | emu (KLEE) | Unit testing | Unit testing (Host / Debug) | Prepare artifacts | Generate documentation | Prepare firmware | Prepare Core artifacts | Publish artifacts | Archive artifacts for Jenkins | Publish documentation | Schedule subprojects | Declarative: Post Actions |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Average stage times: (Average full run time: ~10min 21s) | 3s | 23s | 490ms | 144ms | 30s | 9s | 6s | 6s | 7s | 8s | 6s | 6s | 254ms | 4s | 171ms | 3s | 153ms | 573ms | 285ms | 534ms | 985ms | 193ms | 975ms |
| #220 Dec 16 15:02 1 commit | 1s | 808ms | 505ms | 125ms | 54s | 1min 1s | 1min 1s | 1min 1s | 1min 14s | 1min 18s | 1min 3s | 1min 1s | 326ms | 47s | 160ms | 32s | 94ms | 4s | 201ms | 2s | 7s | 383ms | 1s |
| #219 Dec 16 13:42 1 commit | 842ms | 840ms | 460ms | 112ms | 55s | 28s failed | 172ms failed | 182ms failed | 220ms failed | 127ms failed | 149ms failed | 90ms failed | 354ms | 222ms failed | 187ms | 201ms failed | 192ms | 189ms failed | 324ms | 261ms failed | 321ms failed | 85ms failed | 857ms |
| #218 Dec 16 13:17 1 commit | 1s | 3min 42s | 469ms | 198ms | 55s failed | 184ms failed | 130ms failed | 125ms failed | 137ms failed | 141ms failed | 130ms failed | 180ms failed | 146ms | 155ms failed | 177ms | 131ms failed | 202ms | 194ms failed | 400ms | 293ms failed | 226ms failed | 104ms failed | 938ms |
| #217 Dec 14 19:38 1 commit | 1s | 843ms | 484ms | 84ms | 20s failed | 145ms failed | 181ms failed | 95ms failed | 133ms failed | 142ms failed | 203ms failed | 127ms failed | 250ms | 212ms failed | 106ms | 119ms failed | 83ms | 95ms failed | 272ms | 276ms failed | 258ms failed | 167ms failed | 953ms |
| #216 Dec 08 10:05 1 commit | 1s | 850ms | 535ms | 171ms | 19s failed | 119ms failed | 186ms failed | 100ms failed | 232ms failed | 194ms failed | 208ms failed | 88ms failed | 247ms | 127ms failed | 282ms | 204ms failed | 97ms | 116ms failed | 386ms | 271ms failed | 185ms failed | 220ms failed | 875ms |
| #215 Dec 07 21:19 1 commit | 951ms | 888ms | 465ms | 111ms | 20s failed | 243ms failed | 105ms failed | 147ms failed | 170ms failed | 139ms failed | 183ms failed | 189ms failed | 239ms | 122ms failed | 109ms | 118ms failed | 177ms | 126ms failed | 187ms | 443ms failed | 480ms failed | 186ms failed | 955ms |
| #214 Dec 06 22:20 1 commit | 933ms | 875ms | 436ms | 113ms | 20s failed | 129ms failed | 120ms failed | 152ms failed | 113ms failed | 147ms failed | 182ms failed | 131ms failed | 317ms | 240ms failed | 116ms | 202ms failed | 160ms | 184ms failed | 196ms | 275ms failed | 226ms failed | 186ms failed | 992ms |
| #213 Dec 06 21:39 1 commit | 771ms | 722ms | 612ms | 179ms | 20s failed | 191ms failed | 120ms failed | 203ms failed | 206ms failed | 192ms failed | 196ms failed | 102ms failed | 239ms | 131ms failed | 125ms | 177ms failed | 185ms | 96ms failed | 305ms | 244ms failed | 309ms failed | 207ms failed | 922ms |
| #212 Dec 06 19:24 1 commit | 25s | 1s | 469ms | 177ms | 20s failed | 160ms failed | 120ms failed | 197ms failed | 119ms failed | 196ms failed | 109ms failed | 139ms failed | 255ms | 132ms failed | 237ms | 180ms failed | 195ms | 161ms failed | 219ms | 415ms failed | 404ms failed | 204ms failed | 974ms |
| #211 Nov 25 19:03 1 commit | 2s | 744ms | 471ms | 170ms | 19s failed | 105ms failed | 204ms failed | 131ms failed | 134ms failed | 183ms failed | 154ms failed | 149ms failed | 172ms | 300ms failed | 213ms | 179ms failed | 146ms | 180ms failed | 366ms | 442ms failed | 447ms failed | 191ms failed | 973ms |

# Результаты: автоматизация тестирования и отчетности



# Результаты: автоматизация генерации тестов

```
19  int8_t
20  hb_klee_test(void)
21  {
22      int num;
23  #ifdef HB_KLEE
24      klee_make_symbolic(&num, sizeof(num), "num");
25  #endif
26      printf("%s\n", hb_screen_helper_sprintf_num(num));
27      return 0;
28  }
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ **ТЕРМИНАЛ** ПОРТЫ

```
-476788 044
-786424 000

KLEE: done: total instructions = 117800
KLEE: done: completed paths = 64
KLEE: done: partially completed paths = 0
KLEE: done: generated tests = 64
```

```c
int8_t
hb_klee_test(void)
{
    int num;
#ifdef HB_KLEE
    klee_make_symbolic(&num, sizeof(num), "num");
#endif
    printf("%s\n", hb_screen_helper_sprintf_num(num));
    return 0;
}
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
-3816 090
-6729 598
-6144 000
-8309 008
405217 014
411165 008
817405 664
603261 000
1735425 952
-97012 126
1001106 076
2116183 000
1470057 008
```