Микроконтроллеры STM32

Домашнее задание №8

Евгений Зеленин

27 марта 2025 г.

1. Постановка задачи

Условия задачи.

Создайте проект в среде разработки STM32CubeIDE с имеющейся в наличии платой Nucleo. Подключите библиотеку шифрования соответствующую имеющемуся у вас микроконтроллеру и компилируйте один из представленных примеров. Либо создайте свой пример, опираясь на предоставленные с библиотекой исходники. Целью домашнего задания является: 1. Научиться подключать внешние библиотечные файлы; 2. Получить опыт шифрования данных; 3. Получить опыт работы с предкомпилированными библиотечными файлами. В качестве результата выполнения домашнего задания представьте снимок экрана отображающий в процессе отладки массив с исходными и зашифрованными данными.

2. Подключение и настройка библиотеки

Для выполнения этого задания подключим библиотеку с поддержкой fpu к проекту. Для

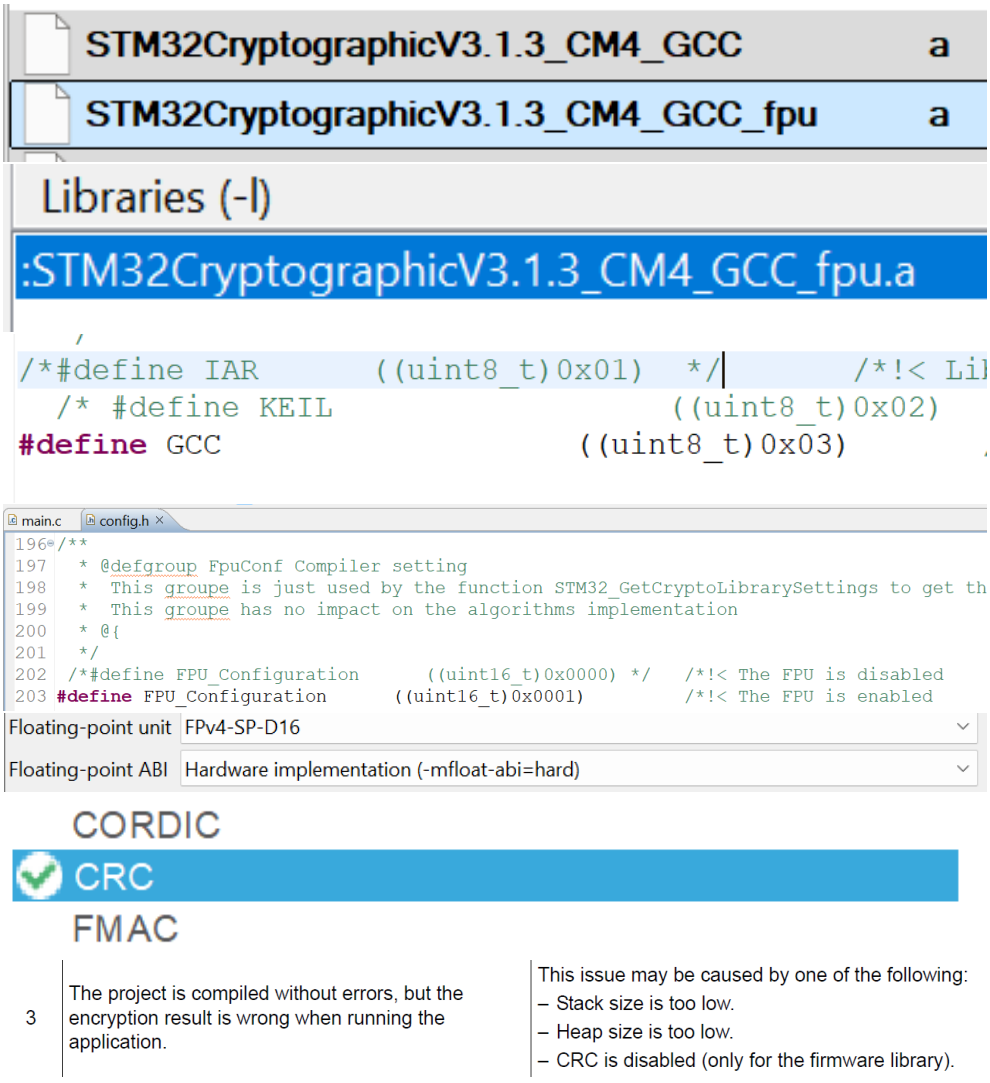

Рис. 1: Подключение и настройка библиотеки

этого скопируем папку STM32_Cryptographic из раздела с библиотекой под конкретный МК

в наш проект. Далее, произведем настройку в соответствии с UM2388 (рисунок 1).,

Стоит отметить, что для корректной работы библиотеки требуется включить CRC, иначе результат расчета алгоритмов будет ошибочным.

Также, в файле config.h требуется произвести настройку библиотеки, а именно: указать версию X.Y.Z, тип компилятора, поддержку FPU, тип процессора, указать с какой оптимизацией собирается проект и т.д.

3. Запуск алгоримов MD5 и SHA1

Для демонстрации работы посчитаем хэш от строки "EvgeniyZelenin"алгоритмами SHA1 и MD5. Опять же, воспользуемся документацией UM2388, где указан подробный порядок действий по использованию функций библиотеки.

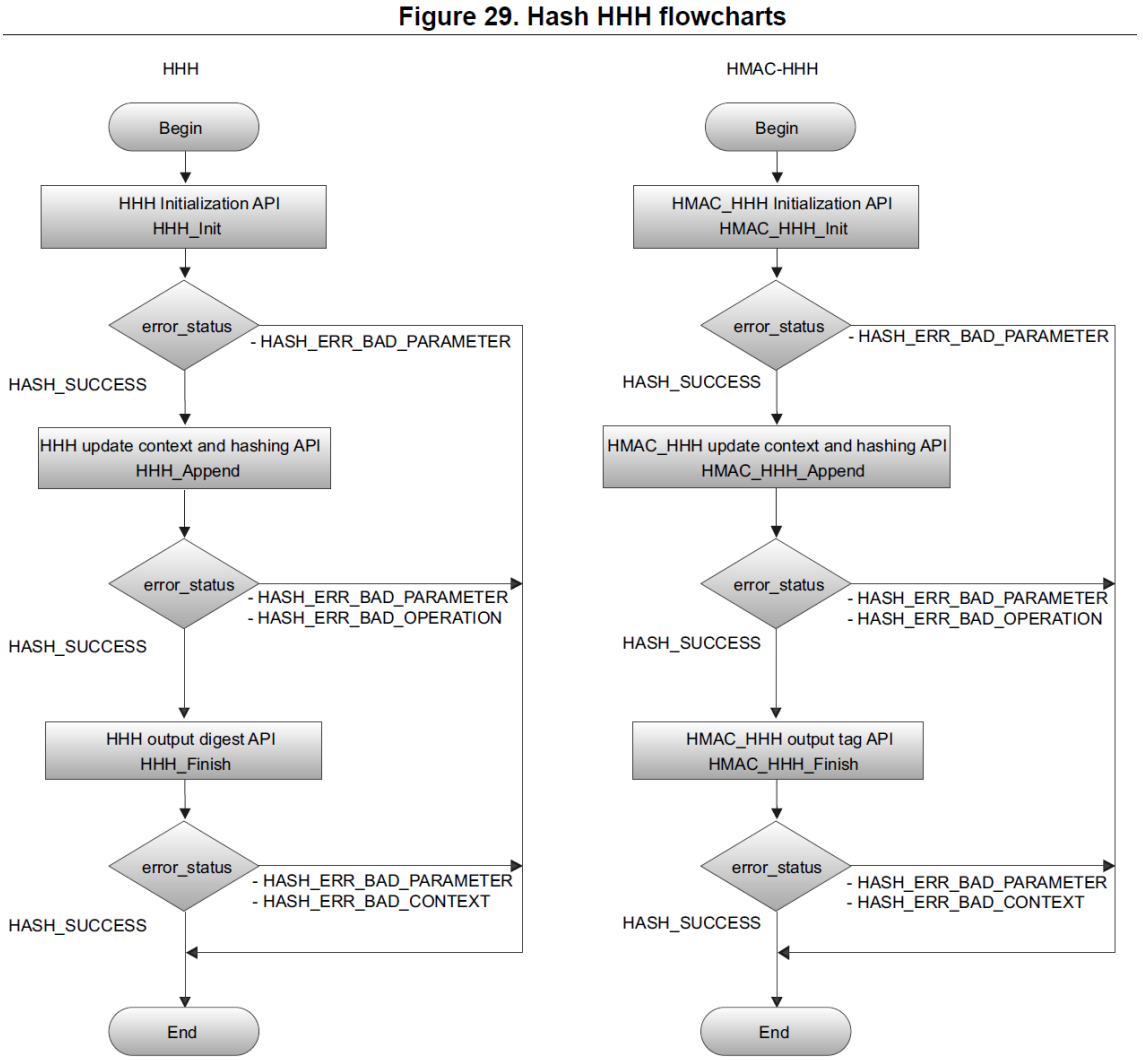

Рис. 2: Алгоритм использования HASH функций

Остается только реализовать эти алгоритмы в коде:

```
/* USER CODE BEGIN WHILE */
```

```
uint8_t input[] = "EvgeniyZelenin";
uint8_t digest[CRL_SHA1_SIZE], digestMD5[CRL_MD5_SIZE];
int32_t outSize, outSizeMD5;
int32_t SHA1_res, MD5_res;

SHA1ctx_stt SHA1ctx_st;
SHA1ctx_st.mTagSize = CRL_SHA1_SIZE;
SHA1ctx_st.mFlags = E_HASH_DEFAULT;
SHA1_res = SHA1_Init(&SHA1ctx_st);
if (SHA1_res == HASH_SUCCESS) {
  SHA1_res = SHA1_Append(&SHA1ctx_st, input, sizeof(input) - 1);
  if (SHA1_res == HASH_SUCCESS) {
    SHA1_res = SHA1_Finish(&SHA1ctx_st, digest, &outSize);
    if (SHA1_res == HASH_SUCCESS) {
      __NOP();
    } else {
      __NOP();
    }
  } else {
    __NOP();
  }
} else {
  __NOP();
}
MD5ctx_stt MD5ctx_st;
MD5ctx_st.mTagSize = CRL_MD5_SIZE;
MD5ctx_st.mFlags = E_HASH_DEFAULT;
MD5_res = MD5_Init(&MD5ctx_st);
if (MD5_res == HASH_SUCCESS) {
  MD5_res = MD5_Append(&MD5ctx_st, input, sizeof(input)-1);
  if (MD5_res == HASH_SUCCESS) {
    MD5_res = MD5_Finish(&MD5ctx_st, digestMD5, &outSizeMD5);
    if (MD5_res == HASH_SUCCESS) {
      __NOP();
    } else {
      __NOP();
    }
  } else {
    __NOP();
  }
} else {
  __NOP();
}
```

Функции NOP() здесь использованы для установки точек останова. После выполнения ал-

горитмов, посмотрим какой результат получился, пересчитаем его с использованием online калькулятора для этой же строки и сравним значения.

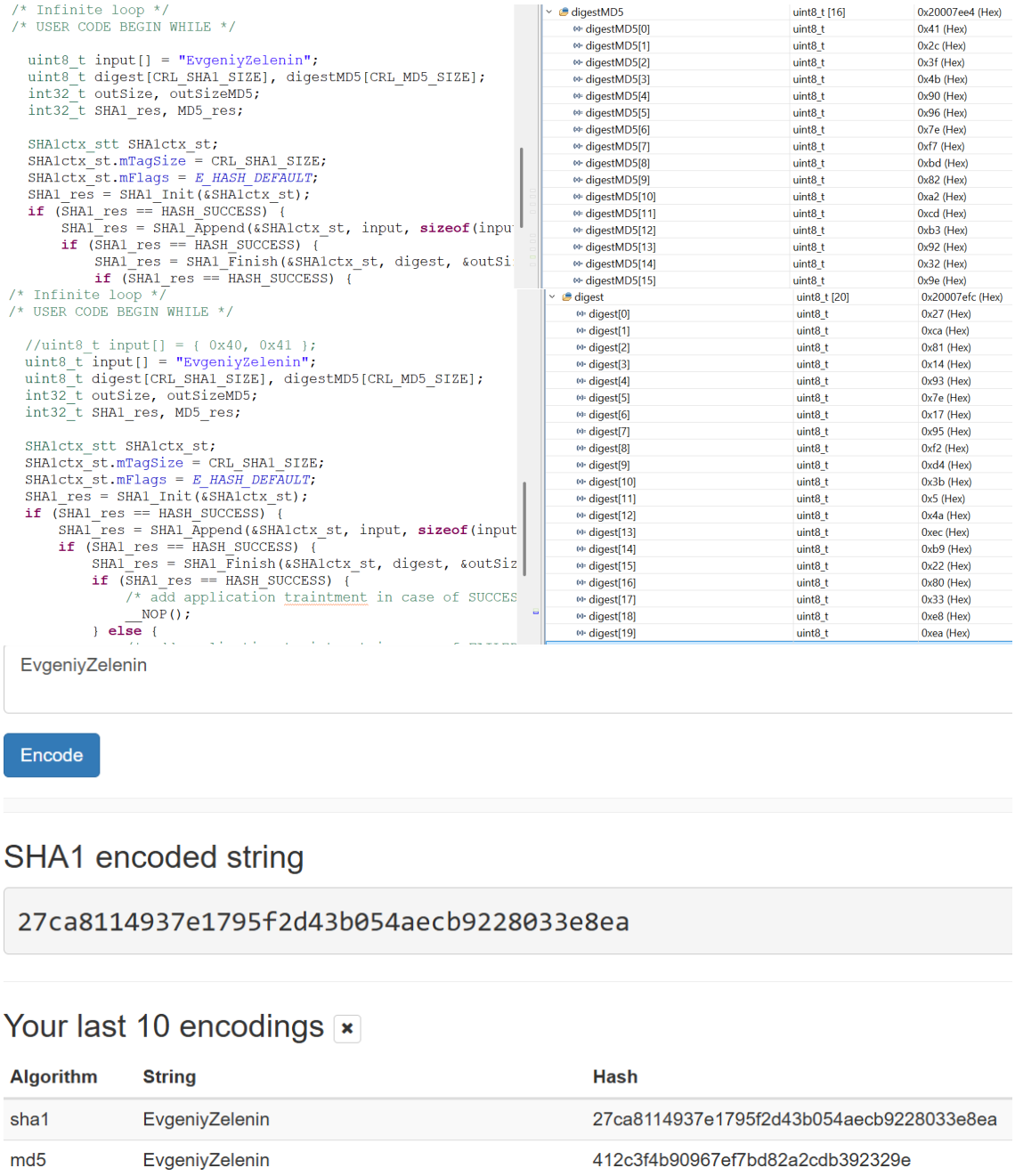

Рис. 3: Сравнение хэшей. Сверху-вниз: MD5, SHA1, online-калькулятор

Как видно из рисунка 3, результаты вычислений совпадают. В материалах к занятию приложен проект и скриншоты.

4. Дополнительные материалы

Материалы к отчету расположены в папке на google диск по следующей ссылке: Материалы к ДЗ №08