2022 국민대학교 정보보안암호수학과 학술동아리 주관 경진대회

5번 문제 - GIFT 경량 암호 최적화 구현 (출제: 동아리 CO2)

GIFT는 2017년에 발표된 경량 블록 암호로, 기존의 PRESENT 블록 암호를 개선한 알고리즘이다. GIFT는 SPN(Substitution-Permutation Network) 구조를 가지며, SubCells, PermBits, AddRoundKey의 세 단계로 이루어진 라운드 함수를 40회 반복하여 암호화한다. 모든 연산이 비트 연산과 Rotation 연산만으로 이루어져 있어 효율적인 암호화가 가능하다. 평문 길이에 따라 GIFT-64와 GIFT-128의 두 가지 버전이 존재한다.

[문제] 8-bit 자료형(unsigned char)만을 사용하여 GIFT-128 암호화 알고리즘을 최적화하여 구현하시오.

[조건]

- 답안 소스 코드는 **C/C++, 어셈블리어**로 작성되어야 한다. (단독 또는 혼용 가능)
- 답안 제출 시, 솔루션 파일(.sln)이 아닌, **코드 파일(.c, .cpp)만 제출**하여야 한다. 필요할 경우, 헤더 파일(.h)을 추가로 제출할 수 있다.
- 적용할 **최적화 방안에 제한은 없으며**, 8-bit 마이크로프로세서 환경에서 구현할 경우 **가산점**을 부여한다.
- 답안 제출 시, **적용한 최적화 방안과 소스 코드에 대한 설명을 포함한 PDF 문서를 함께 제출**하여야 한다.
- 답안 제출 시, 작성한 코드의 성능 측정이 이루어져야 하며, **아래와 같은 성능 측정 함수를 작성하여 측정 결과를 제출하시오**. 단, MacOS에서는 정상적으로 작동하지 않을 수 있으니 MacOS에서 구현한 경우, 암호화 함수만 작성하여 제출하시오.

```c
#include <stdio.h>
#include <intrin.h>

#define ITERNUM 100000

__int64 cpucycles(){
    return __rdtsc();
}

void performance_calc(){
    unsigned long long start, end;
    start = cpucycles();
    for(int i = 0; i < ITERNUM; i++){
        // 암호화 함수 호출
    }
    end = cpucycles();
    printf("Encryption rdtsc = %10lld\n", (end - start) / ITERNUM);
}
```

[유의사항]

- 모든 코드는 정상적으로 컴파일이 되어야 하며, **컴파일이 되지 않는 코드는 채점하지 않음**.
- Test Vector가 틀린 경우, **구현 방식에 관계없이 감점 처리함**.
- **코드 카피가 적발될 경우 0점 처리함**.

[참고 자료]

GIFT-COFB 레퍼런스 문서:
https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-submissions/gift-cofb.zip

● Implementation->crypto_aead->giftcofb128v1->ref 폴더 안의 gift128.c와 gift128.h 코드는 레퍼런스 코드입니다. 해당 코드를 그대로 사용하는 것은 지양해 주시기 바랍니다.
● 레퍼런스 문서는 Documents 폴더 안의 GIFT-COFB_v1.1.pdf 파일입니다. 해당 문서의 2.4절을 참고하시기 바랍니다.

[8-bit 마이크로프로세서 환경 설정]

개발환경은 Microchip 사에서 무료로 제공하는 개발 툴인 Microchip Studio를 사용 가능하다.

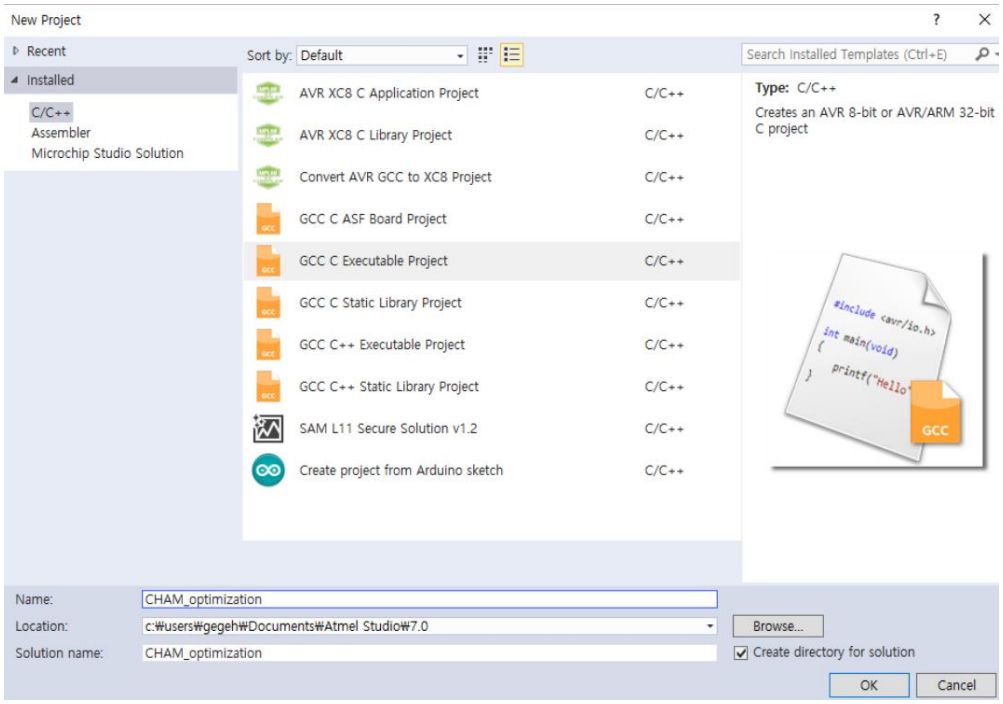
- 다운로드 URL:
https://www.microchip.com/en-us/tools-resources/develop/microchip-studio#Downloads

- Getting Started with MicroChip Studio:
https://www.youtube.com/watch?v=pbThbRL7UGM&list=PLtQdQmNK_0DQZjOQiqLyJwUcc88MJPOiD

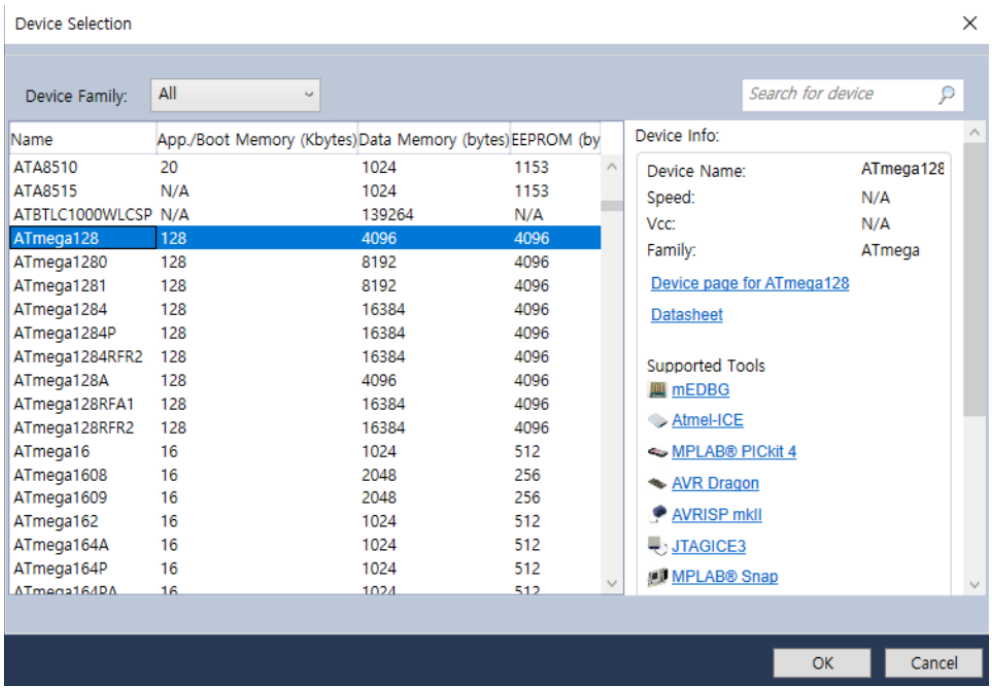
- AVR assembly 명령어:
http://ww1.microchip.com/downloads/en/devicedoc/atmel-0856-avr-instruction-set-manual.pdf

[프로젝트 생성]

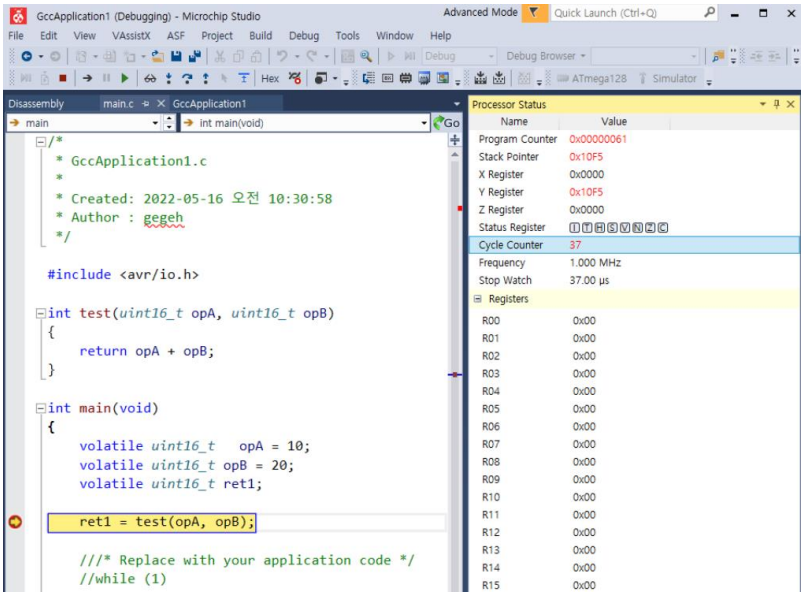
- Microchip studio 실행File 메뉴→New→Project→GCC Executable Project 선택



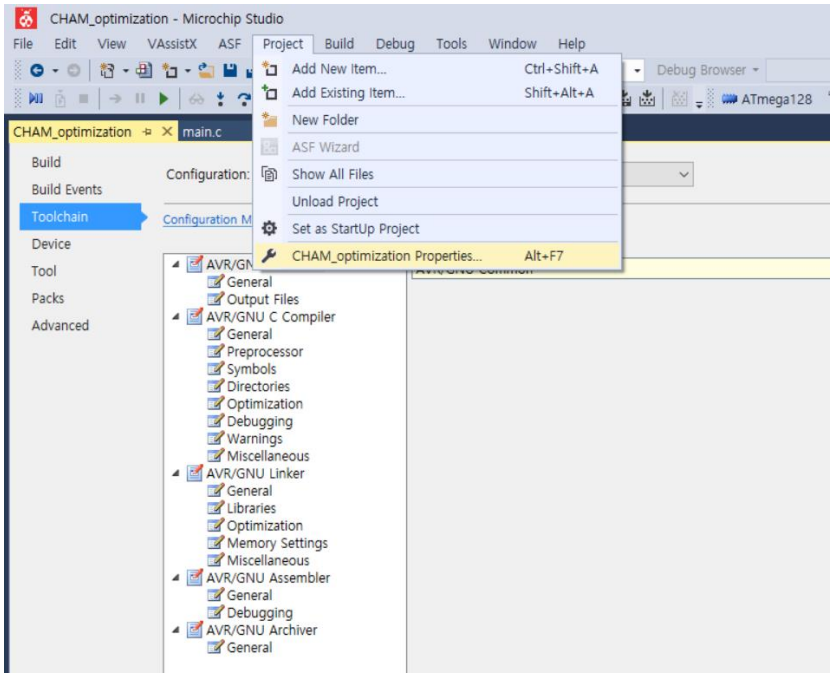
- Device selection에서 ATmega128 MCU 선택



- Project 메뉴에서 컴파일 옵션 설정 가능 (예: AVR/GNU C Compiler- Optimization). 최적화 레벨은 O2로 한다.



- 소스 코드 작성 및 빌드 후, 디버그를 진행하여 시뮬레이터 상에서 Clock Cycle 측정 가능 (Debug 메뉴-Windows-Processor Status에서 Cycle Counter 확인)



- Microchip 사용 매뉴얼:
http://ww1.microchip.com/downloads/en/devicedoc/atmel-42167-atmel-studio_user%20guide.pdf

문의 : mike0726@kookmin.ac.kr