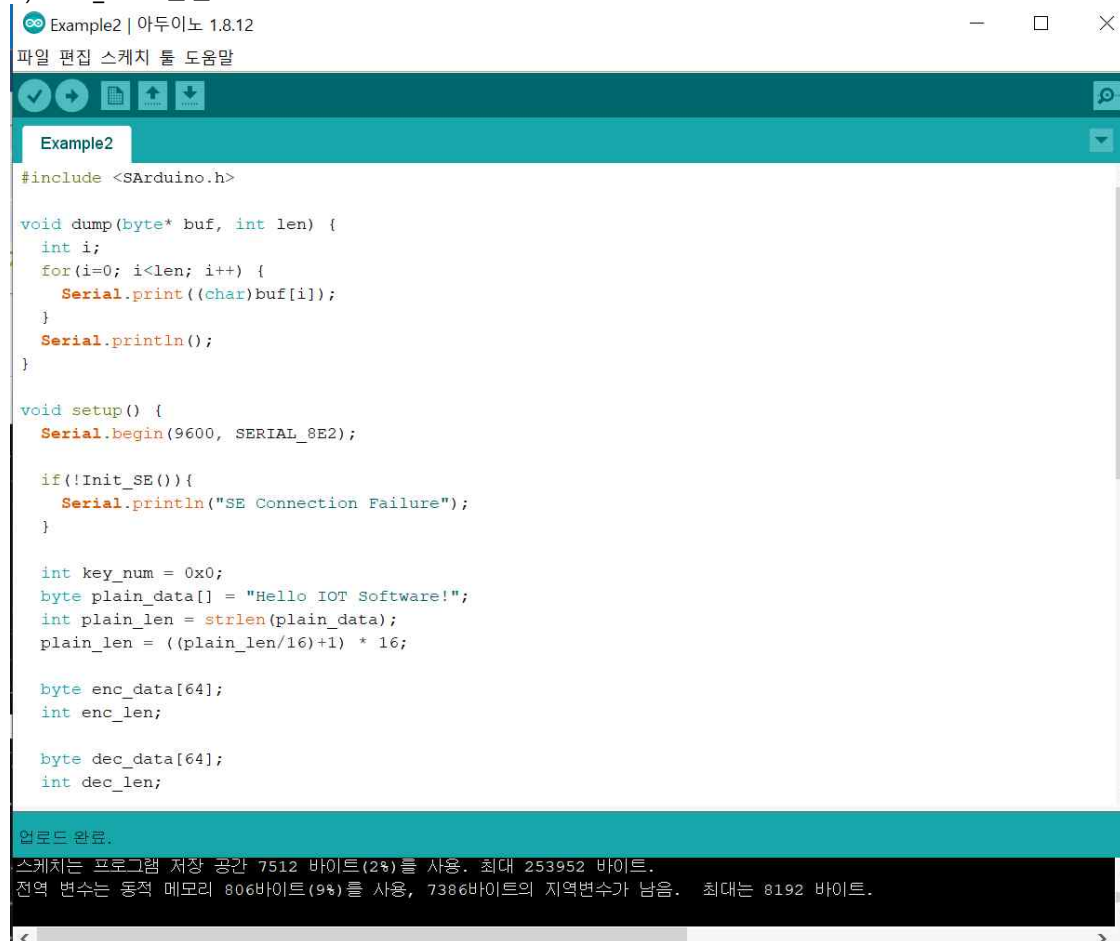


```
COM3  
|  
  
plain data: Hello IOT Software!  
digest: 89f▲ssssss+sj;Jss|sss55s▲s↑cs
```

Hash function으로 암호화된 digest 확인하였습니다.

2) AES_128 실습



```
#include <Arduino.h>

void dump(byte* buf, int len) {
  int i;
  for(i=0; i<len; i++) {
    Serial.print((char)buf[i]);
  }
  Serial.println();
}

void setup() {
  Serial.begin(9600, SERIAL_8E2);

  if(!Init_SE()){
    Serial.println("SE Connection Failure");
  }

  int key_num = 0x0;
  byte plain_data[] = "Hello IOT Software!";
  int plain_len = strlen(plain_data);
  plain_len = ((plain_len/16)+1) * 16;

  byte enc_data[64];
  int enc_len;

  byte dec_data[64];
  int dec_len;
}
```

업로드 완료.

스케치는 프로그램 저장 공간 7512 바이트(2%)를 사용. 최대 253952 바이트.
전역 변수는 동적 메모리 806바이트(9%)를 사용, 7386바이트의 지역변수가 남음. 최대는 8192 바이트.

COM3

```
plain data: Hello IOT Software!^_||??>??r
enc_data: {K?!!$d^$??D???'Mjxj`$sG?j??>r?P?sa?rJ??q??Q?V?
dec_data: Hello IOT Software!
```

Plain data의 평문이 encapsulation 되었다가 다시 복조 후 그대로 나오는 것을 확인하였습니다.

3) SHA_256 실습

Example3 | 아두이노 1.8.12

파일 편집 스케치 툴 도움말



Example3

```
}

int key_num = 0x0;
byte plain_data[64] = "Hello IOT Software!";
int plain_len = strlen(plain_data);

byte enc_data[128];
int enc_len;

byte dec_data[64];
int dec_len;

if(!Generate_RSA1024Key(key_num))
    Serial.println("Set RSA1024 Key Pair Failure");

Serial.print("plain data: ");
dump(plain_data, plain_len);

if(Encrypt_RSA1024(key_num, plain_data, plain_len, enc_data, &enc_len)) {
    Serial.print("enc_data: ");
    dump(enc_data, enc_len);
}

else
    Serial.println("Encrypt plain_data Failure");

if(Decrypt_RSA1024(key_num, enc_data, enc_len, dec_data, &dec_len)) {
    Serial.print("dec data: ");
```

COM3

```
plain data: Hello IOT Software!
enc_data: 1+7.7.5'Q&5↑~{Q\||5'5,5^5_T].5-
dec_data: Hello IOT Software!
```

개인 Private key로 복조하는 것을 확인하였습니다.

실습과제)

우선 SHA-256 으로 해당 데이터를 해쉬화 하여 암호화 시켰습니다.

그 다음 RSA-1024 통하여 그 해쉬 데이터를 보안 해제 시키는데, 검증을 위해서 FOR문을 통해 digest[i] 와 dec_data[i]가 같은지 확인하였습니다. 바뀐 Library가 아닌 원래 있었던 Library를 통해 실습을 진행하였기 때문에 Library안에 있는 Verify 함수를 쓰지 않고

digest 와 dec_Data를 비교하며 검증하였습니다. 아래는 코드와 실행 결과입니다.

```
Assignment
#include <Arduino.h>

#define PRIVATE 0
#define PUBLIC 1

void dump(byte* buf, int len) {
    int i;
    for(i=0; i<len; i++) {
        Serial.print((char)buf[i]);
    }
    Serial.println();
}

void setup() {
    Serial.begin(9600, SERIAL_8E2);

    if(!Init_SE()){
        Serial.println("SE Connection Failure");
    }

    int key_num = 0x0;
    int verification = 1;

    byte plain_data[] = "나는 2018년 6월 1일에 홍길동에게 100만원을 입금하였다";
    int plain_len = strlen(plain_data);

    byte digest[32];
    int digest_len = 32;

    byte enc_data[128];
    int enc_len;

    byte dec_data[64];
    int dec_len;

    Serial.print("plain data: ");
    dump(plain_data, plain_len);

    if(SHA_256(plain_data, plain_len, digest, &digest_len)) {
        Serial.print("digest: ");
        dump(digest, digest_len);
    }
}
```

```

else {
    Serial.println("SHA_256 Failure");
}

if(!Generate_RSA1024Key(key_num))
    Serial.println("Set RSA1024 Key Pair Failure");

if(Encrypt_RSA1024(key_num, PUBLIC, digest, digest_len, enc_data, &enc_len)) {
    Serial.print("enc_data: ");
    dump(enc_data, enc_len);
}

else
    Serial.println("Encrypt plain_data Failure");

if(Decrypt_RSA1024(key_num, PRIVATE, enc_data, enc_len, dec_data, &dec_len)) {
    Serial.print("dec_data: ");
    dump(dec_data, dec_len);
}

else
    Serial.println("Decrpyt enc_data Failure");

for(int i=0; i<dec_len; i++) {
    if(digest[i] != dec_data[i]) {
        Serial.println("Verification Fail");
        verification = 0;
        break;
    }
}

if(verification == 1)
    Serial.println("Veiricaion Success!");
}

void loop() {
}

```

SE Connection Success

plain data: 나는 2018년 6월 1일에 홍길동에게 100만원을 입금하였다

digest: -è?@???s?`q??碯!RYP????-?

enc_data: tm\$]_?(l(6?L⁺Raw???i?L□%??4??L???y?%?o??????1?,?W?Q|?←<|m??4
?N>+E?j?→?!!??w?T?,*?@?^c?l??≤=,D! "SM/?-?→*???d?L???W□Ak'

dec_data: -è?@???s?`q??碯!RYP????-?

Verification Success

Verification Success 문자 출력을 통해 복조된 것을 확인하였습니다.