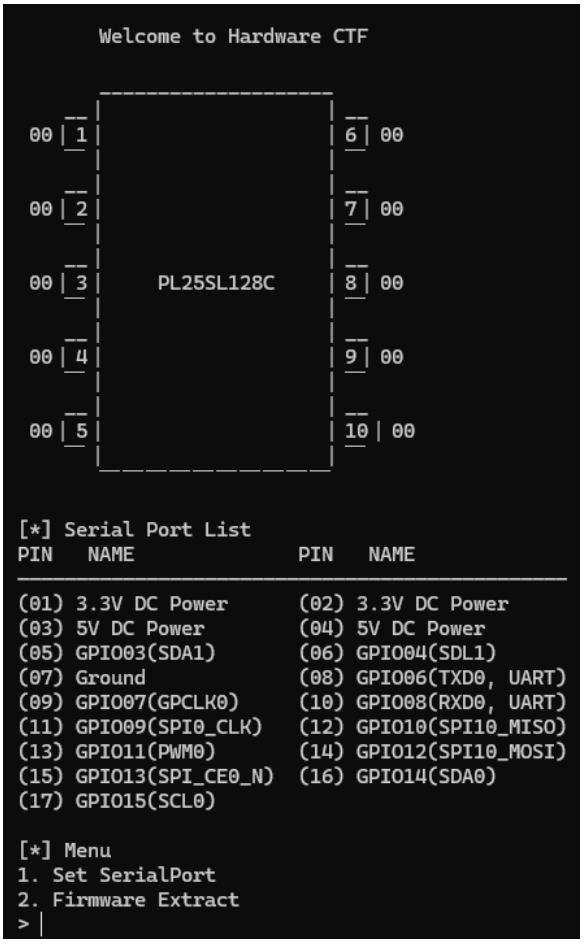
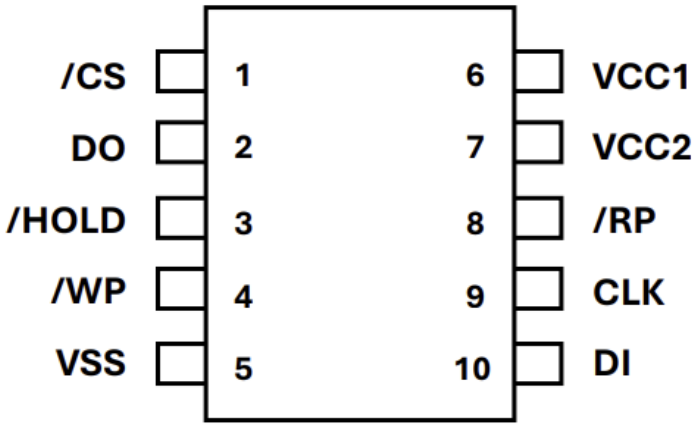
Write-up

1) Flash Memory Firmware Extract



문제에서 제공되는 nc를 붙어보면, 위와 같이 출력창을 볼 수 있다.

Figure.1 CONNECTION DIAGRAMS



10 - LEAD SOP

Table 1. Pin Names

Index	PIN NAME	FUNCTION	TYPE
1	/CS	Chip Select Input	I
2	DO	Data Output	O
3	/HOLD	Hold Input	I
4	/WP	Write Protect Input	I
5	VSS	Ground	P
6	VCC1	3.3V Power Supply	P
7	VCC2	5V Power Supply	P
8	/RP	Read Protect Input	I
9	CLK	Serial Clock Input	I
10	DI	Data Input	I

문제에서 제공되는 데이터 시트를 읽고, Serial Port List와 적절히 매핑해준다.

Table 2. Instruction Set

Instruction Name	Byte 1 Code	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
Read Data	03h	A23-A16	A15-A8	A7-A0	(D7-D0)	(Next byte)	continuous
Read Identification	9Fh	(M7-M0)	(ID15-ID8)	(ID7-ID0)	(1)		
Manufacturer/Device ID	90h	dummy	dummy	00h	(M7-M0)	(ID7-ID0)	(2)
Deep Power-down	B9h						
Release from Deep Power-down, and Device ID	ABh	dummy	dummy	dummy	(ID7-ID0)		

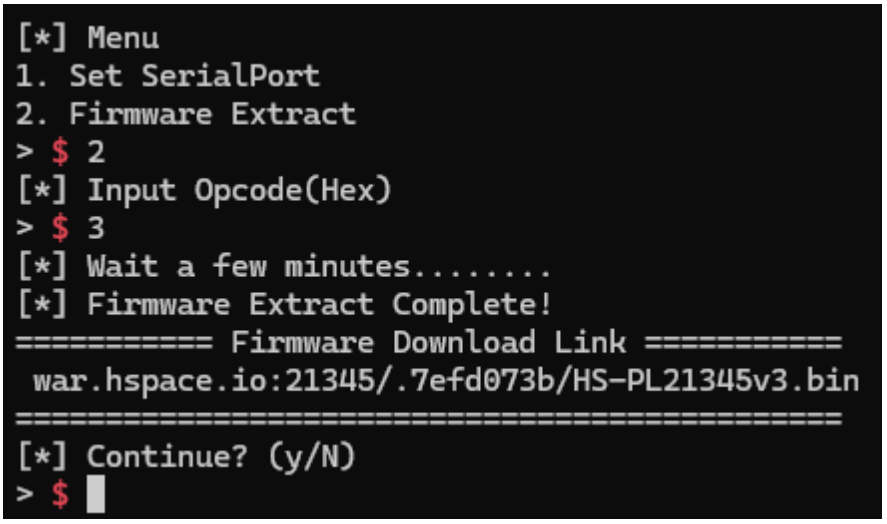
1. (M7-M0) : Manufacturer, (ID15-ID8) : Memory Type, (ID7-ID0) : Memory Capacity
2. The Manufacturer ID and Device ID bytes will repeat continuously until CS# terminates the instruction.

이후 Instruction Set에 있는 Read Data(0x3)를 입력해서 펌웨어를 읽는다.

추출 코드는 아래와 같다.

```
from pwn import *

p = process('./prob')
#p = remote('127.0.0.1', 21346)

CS = 1
DO = 2
HOLD = 3
WP = 4
VSS = 5
VCC1 = 6
VCC2 = 7
RP = 8
CLK = 9
DI = 10

def set_port(index, serial):
    p.sendlineafter(b'> ', b'1')
    p.sendlineafter(b'> ', str(index).encode())
    p.sendlineafter(b'> ', str(serial).encode())

set_port(CS, 15)
set_port(DO, 12)
```

```
set_port(VSS, 7)
set_port(VCC1, 2)
set_port(VCC2, 4)
set_port(RP, 3)
set_port(CLK, 11)
set_port(DI, 14)

p.sendlineafter(b'> ', b'2')
p.sendlineafter(b'> ', b'3')
p.sendlineafter(b'> ', b'y')

p.interactive()
```



추출이 완료되면, 서버 내부 저장해놓은 펌웨어 파일을 얻을 수 있다.

2) 펌웨어 분석

```
binwalk HS-PL21345v3.bin
```

추출한 펌웨어를 binwalk를 돌리고 문제에서 제공된 웹 서비스의 폴더를 들어가보면 아래와 같은 파일 목록을 볼 수 있다.

```
yejun@LEE:~/physicallab/hardwareCTF/firmware-user/squashfs-root/web/cgi-bin$ ls
fd2895412cffbc6a
yejun@LEE:~/physicallab/hardwareCTF/firmware-user/squashfs-root/web/cgi-bin$ file fd2895412cffbc6a
fd2895412cffbc6a: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), statically linked, BuildID[sha1]=e9ef6f3037fd5
ccc5e47a29c73d0d22b3bc588e2, for GNU/Linux 3.2.0, not stripped
```

cgi-bin 경로에 바이너리 형태의 수상한 파일을 발견.

3) fd2895412cffbc6a 바이너리 분석

main 함수

```
16  v10 = getenv("QUERY_STRING");
17  if ( v10 && strlen(v10) > 4 && !strncmp(v10, "cmd=", 4) )
18  {
19    snprintf(&v12, 0x100, "%s", (v10 + 4));
20    v8 = 0;
21    v9 = 0;
22    while ( v13[v8 - 4] )
23    {
24      if ( v13[v8 - 4] == 0x2B )
25      {
26        v13[v9 - 4] = 0x20;
27      }
28      else if ( v13[v8 - 4] == 0x25
29             && (*(*_ctype_b_loc() + 2 * v13[v8 - 3]) & 0x1000) != 0
30             && (*(*_ctype_b_loc() + 2 * v13[v8 - 2]) & 0x1000) != 0 )
31      {
32        v11[0] = v13[v8 - 3];
33        v11[1] = v13[v8 - 2];
34        v11[2] = 0;
35        v6 = strtol(v11, 0, 0x10);
36        v5 = v6;
37        v13[v9 - 4] = v6;
38        v8 += 2;
39      }
40      else
41      {
42        v5 = v13[v8 - 4];
43        v13[v9 - 4] = v13[v8 - 4];
44      }
45      ++v8;
46      ++v9;
47    }
48    command_result(&v12, v5, &v12);
```

cmd 파라미터를 가져와 command_result 함수로 넘긴다.

command_result() 함수

```
39  }
40  if ( *(a1 + 77) == 2 )
41    flag(v8, v9, v10);
42  printf("</pre>");
43  return printf("</body></html>");
44 }
00000778 command_result:40 (10778)
```

앞서 main에서 넘긴 v12 주소 + 77 의 값이 2라면, flag() 함수가 실행된다.

flag() 함수

```
1  int __fastcall flag(int a1, int a2, int a3)
2  {
3    int v3; // r1
4    int v4; // r2
5    int v6; // r1
6    int v7; // r2
7    int v8; // r3
8    int v9; // r1
9    int v10; // r2
10   int v11; // r1
11   int v12; // r2
12   int v13; // r3
13   int v14; // [sp+0h] [bp-40Ch]
14   char v15[8]; // [sp+4h] [bp-408h] BYREF
15
16   v14 = fopen("flag", &unk_70D98, a3, 0);
17   if ( !v14 )
18     return puts("Failed to open the flag file.", v3, v4, 0);
19   puts("<h2>flag</h2>", v3, v4, v14);
20   puts("<pre>", v6, v7, v8);
21   while ( fgets(v15, 0x400, v14) )
22     printf("%s", v15);
23   puts("</pre>", v9, v10, 0);
24   v13 = ferror(v14);
25   if ( v13 )
26     puts("Error reading from file.", v11, v12, v13);
27   return fclose(v14);
28 }
```

flag 함수는 flag 파일을 fopen으로 읽어주는 함수이다.

flag를 얻기 위한 웹 서버 요청은 다음과 같다.

```
http://[url]:[port]/cgi-bin/webshell?cmd=aaaaaaaaaaaaaaaaaaaa
```

4) FLAG

```
FLAG{h5pAc3_phY5ica1LaB_CTF_easy_c1eAr!@}
```