

Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

Mibspi(Master_Slave)

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

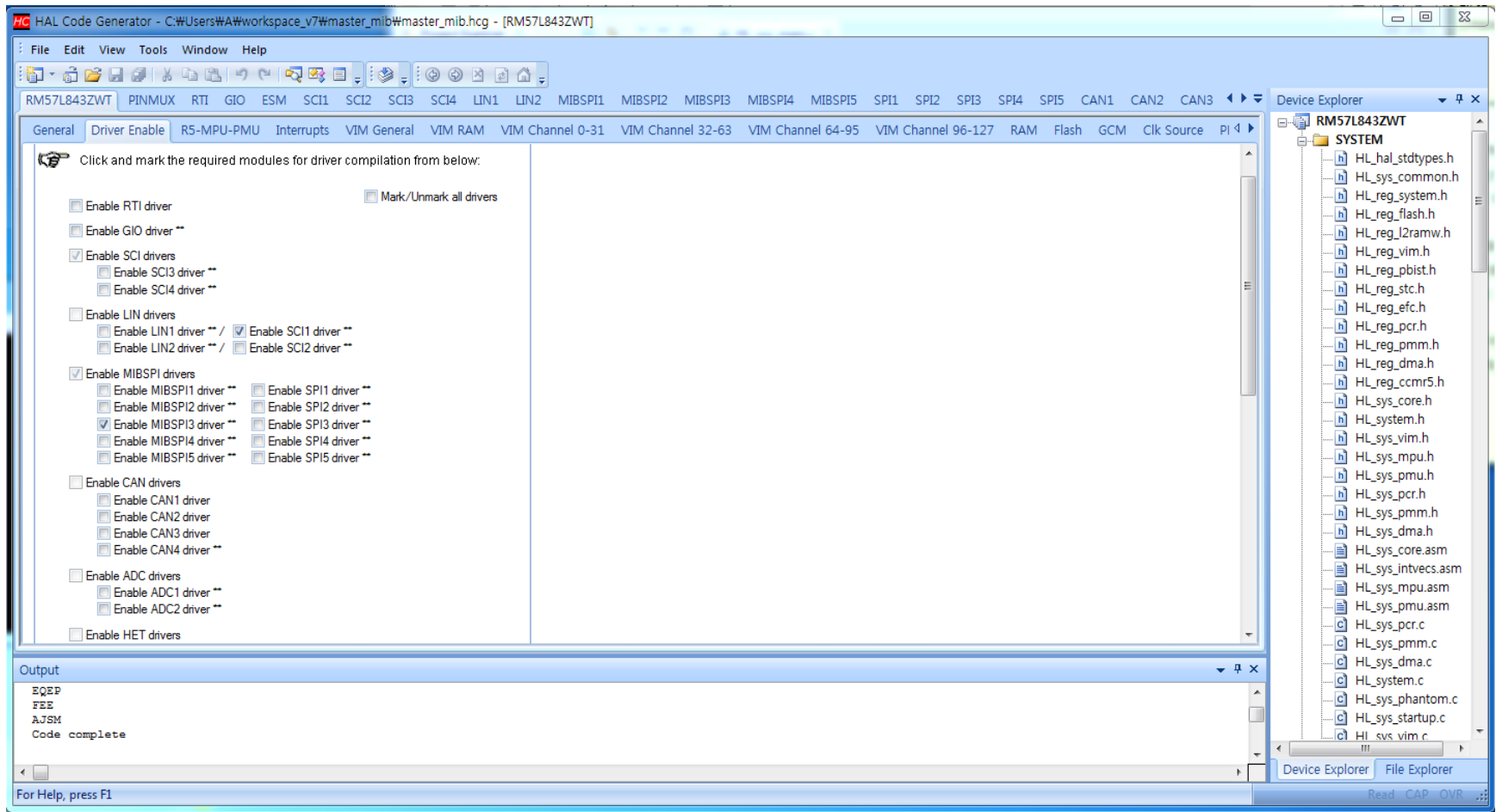
학생 – 변진혁

xollgun@gmail.com

학생 – 김형준

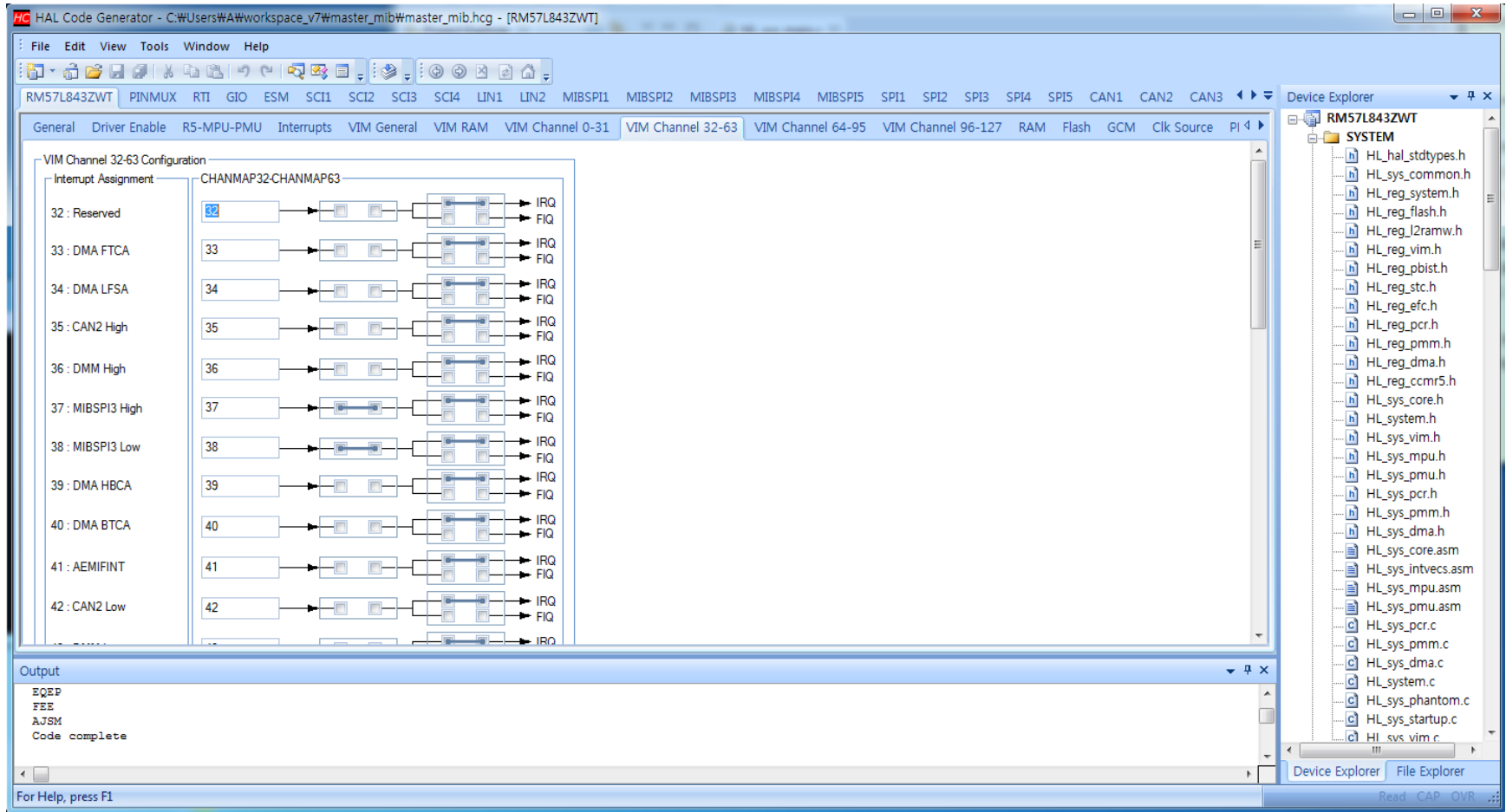
kimdj417@gmail.com

1. Halcogen Setting (Master, Slave)



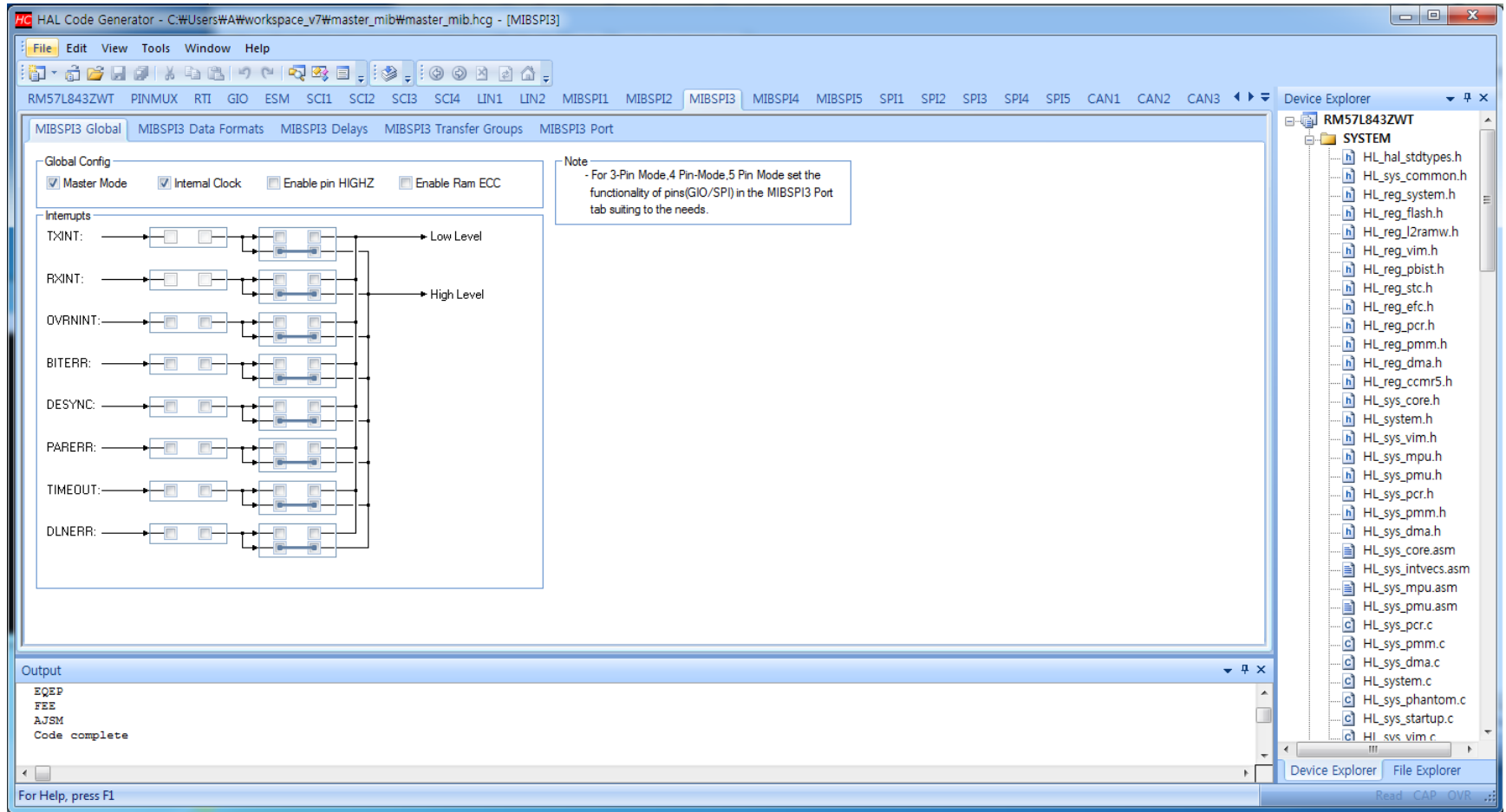
Master와 Slave와 공통적으로 Setting해주면 된다.
(SCI 1번과 MIBSPI3을 연결해준다.)

2.Halcogen Setting(Master,Slave)



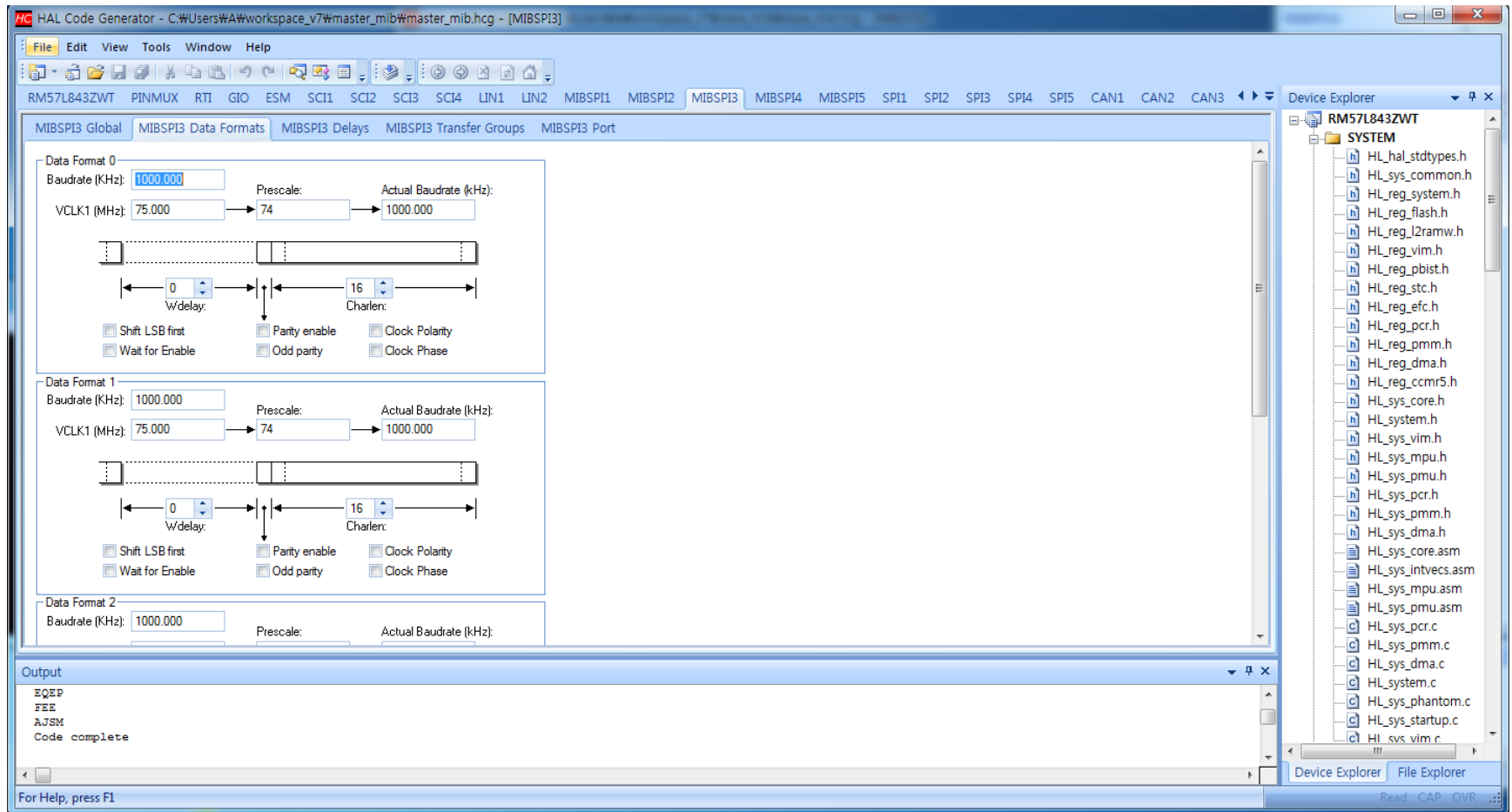
Master와 Slave와 공통적으로 Setting해주면 된다.
(VIM 채널(32-43)에 MIBSPI3 High/Low를 연결해준다.)

3.Halcogen Setting(Master)



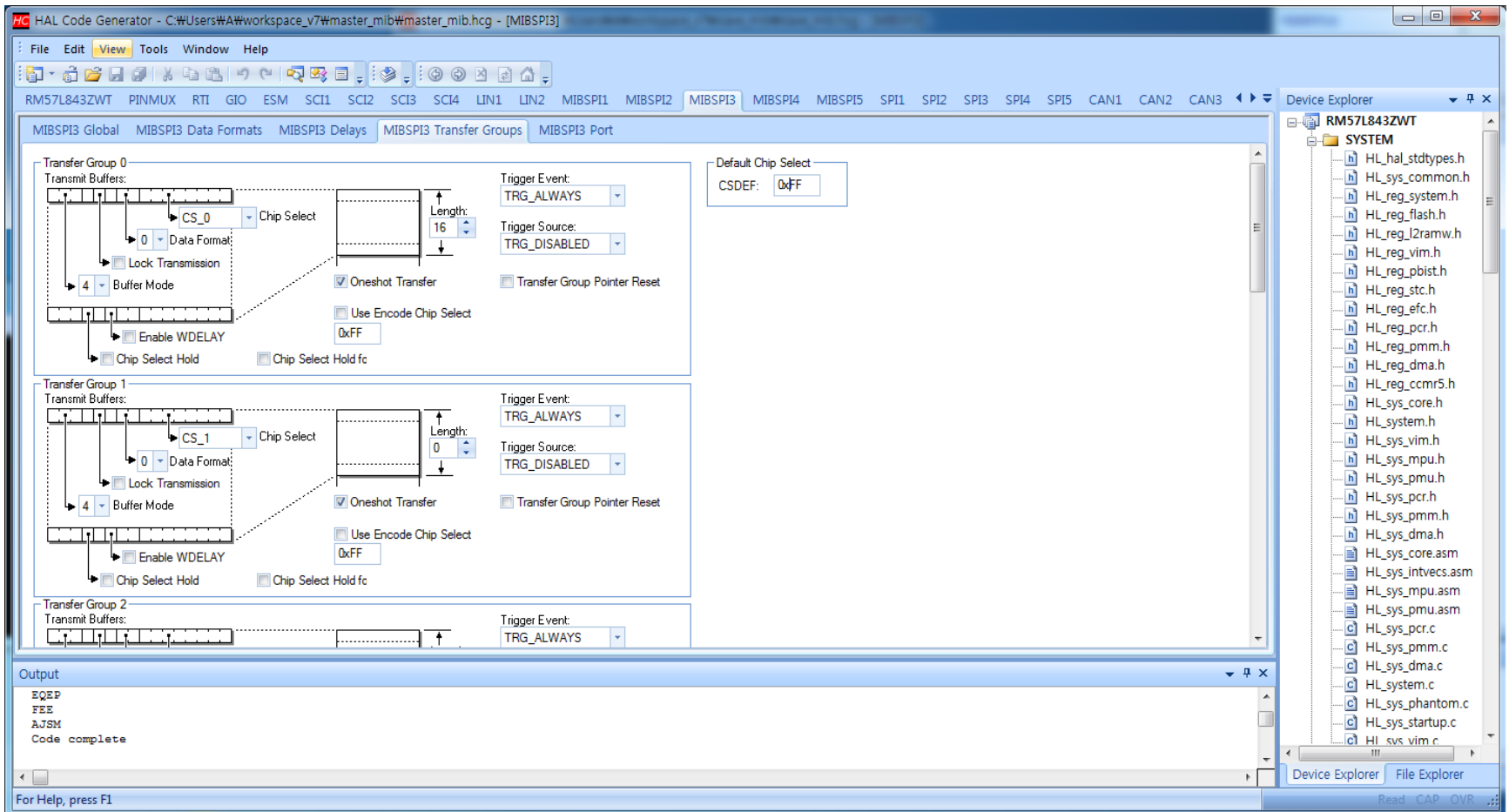
Master만 해당 하는 Setting으로 Master Mode와 internal Clock을 연결해준다.

4. Halcogen Setting (Master, Slave)



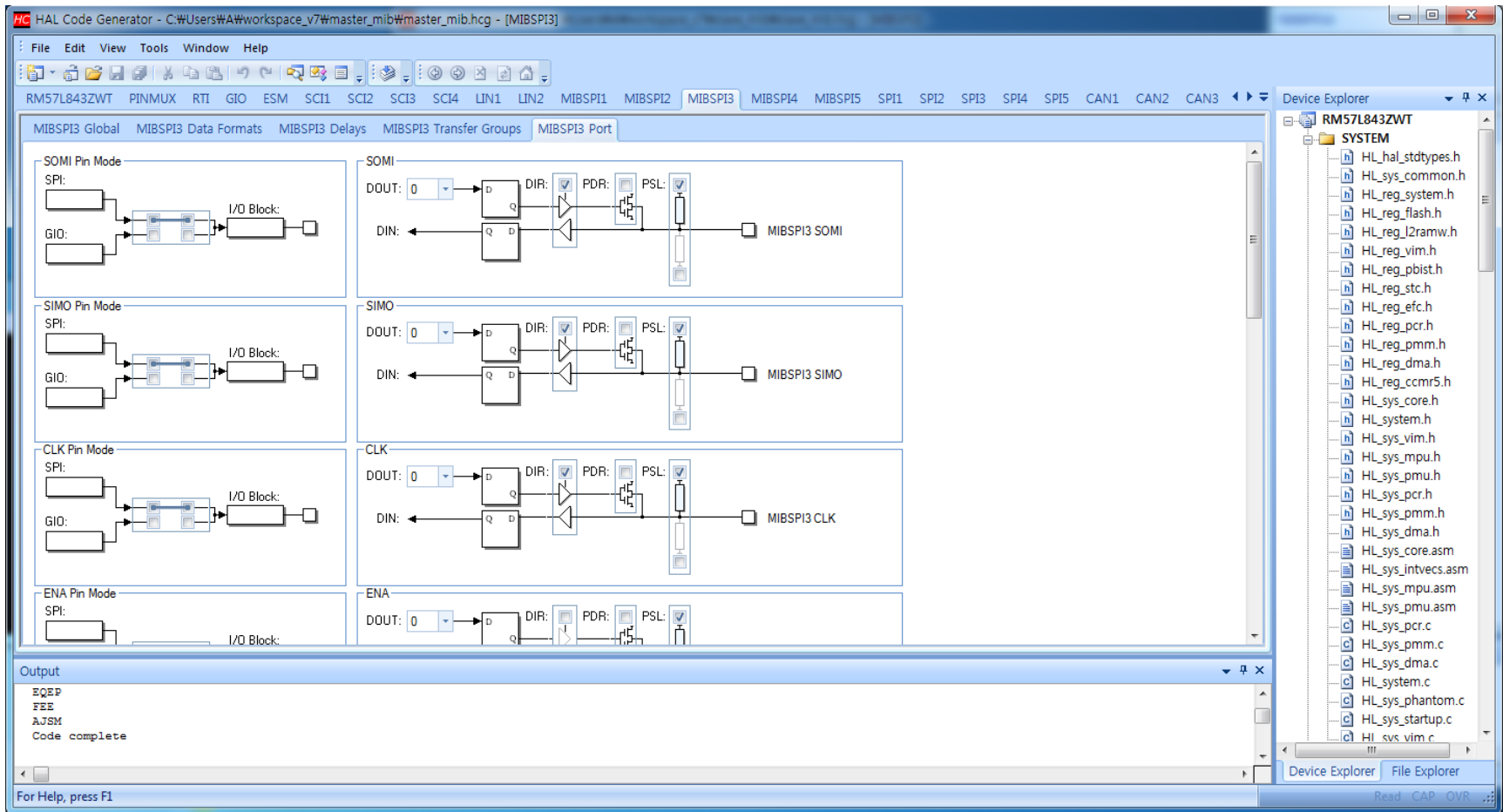
1000KHz로 맞춰주고 데이터의 길이를 16으로 맞춰준다..

5. Halcogen Setting (Master, Slave)



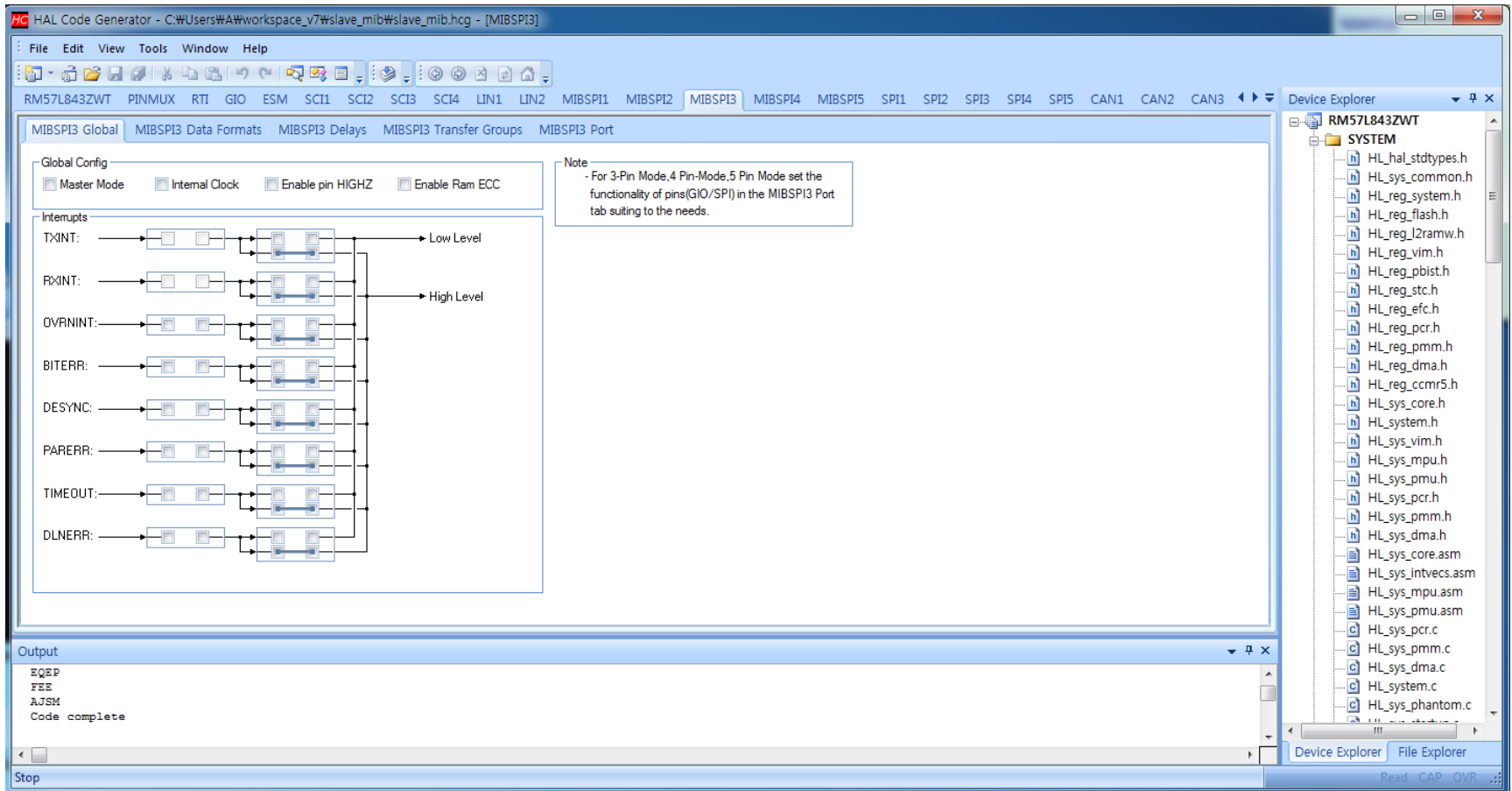
Transfer 0그룹만 사용하기 때문에 0번그룹에 있는 Length 부분을 16으로 setting 해준다.

6. Halcogen Setting (Master, Slave)



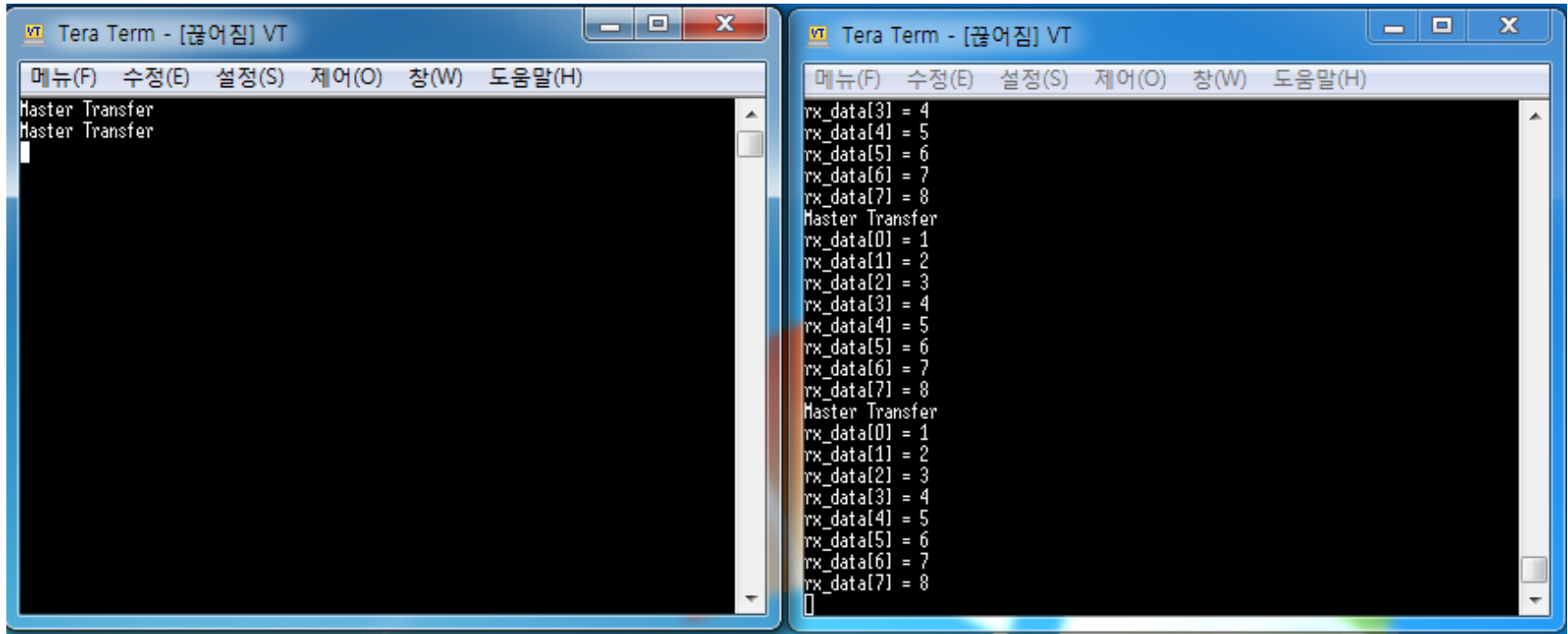
SOMI, SIMO, CLK, CS 전부 활성화를 해준다.

7. Halcogen Setting(Slave)



Slave Setting에서 Master Mode와 internal Clock을 해제시킨다.

8.Tera Term(Master,Slave)



```
VT Tera Term - [끝어짐] VT
메뉴(F) 수정(E) 설정(S) 제어(O) 창(W) 도움말(H)
Master Transfer
Master Transfer

VT Tera Term - [끝어짐] VT
메뉴(F) 수정(E) 설정(S) 제어(O) 창(W) 도움말(H)
rx_data[3] = 4
rx_data[4] = 5
rx_data[5] = 6
rx_data[6] = 7
rx_data[7] = 8
Master Transfer
rx_data[0] = 1
rx_data[1] = 2
rx_data[2] = 3
rx_data[3] = 4
rx_data[4] = 5
rx_data[5] = 6
rx_data[6] = 7
rx_data[7] = 8
Master Transfer
rx_data[0] = 1
rx_data[1] = 2
rx_data[2] = 3
rx_data[3] = 4
rx_data[4] = 5
rx_data[5] = 6
rx_data[6] = 7
rx_data[7] = 8
```

양방향 통신이 된다. 나 같은 경우 Slave Port에서 CLK핀을 Enable 안해서 Slave 데이터가 Master로 안 넘어 간다. 그것만 Enable해주면 정상적으로 작동할 듯 하다.

9.Code(Master,Slave)

```
1 #include "HL_sci.h"
2 #include "HL_mibspi.h"
3 #include "HL_sys_common.h"
4 #include "HL_system.h"
5 #include <stdio.h>
6 #include <string.h>
7
8 #define UART    sciREG1
9
10 uint16 tx_data[8] = {1,2,3,4,5,6,7,8};
11 uint16 rx_data[8] = {0};
12
13 void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 len);
14
15 void main(void)
16 {
17     volatile i;
18     char txt_buf[256];
19     unsigned int buf_len;|
20
21     _enable_IRQ_interrupt_();
22
23     sciInit();
24     mibspiInit();
25
26     mibspiEnableGroupNotification(mibspiREG3, 0, 1);
27     mibspiSetData(mibspiREG3, 0, &tx_data[0]);
28     mibspiTransfer(mibspiREG3, 0);
```

Master와 Slave와 공통되게 사용하면 된다. 단 정확히 통신이 되는지 확인하기 위해 tx_data 부분을 Slave에서는 다른 값으로 바꾸어 사용하자.

10.Code(Master,Slave)

```
30  sprintf(txt_buf, "Master Transfer\n\r\0");
31  buf_len = strlen(txt_buf);
32  sciDisplayText(sciREG1, (uint8 *)txt_buf, buf_len);
33
34  while(!(mibspiIsTransferComplete(mibspiREG3, 0)))
35  {
36      ;
37  }
38  mibspiGetData(mibspiREG3, 0, &rx_data[0]);
39
40  for(i=0;i<8;i++)
41  {
42      sprintf(txt_buf, "rx_data[%d] = %d\n\r\0", i, (char)rx_data[i]);
43      buf_len = strlen(txt_buf);
44      sciDisplayText(sciREG1, (uint8 *)txt_buf, buf_len);
45  }
46
47  while(1);
48
49
50
51 }
52
```

Master와 Slave와 공통되게 사용하면 된다.

11.Code(Master,Slave)

```
53 void sciDisplayText(sciBASE_t *sci, uint8 *text,uint32 length)
54 {
55     while(length--)
56     {
57         while ((UART->FLR & 0x4) == 4);
58         sciSendByte(UART, *text++);
59     };
60 }
61
62
63
```

Master와 Slave와 공통되게 사용하면 된다.