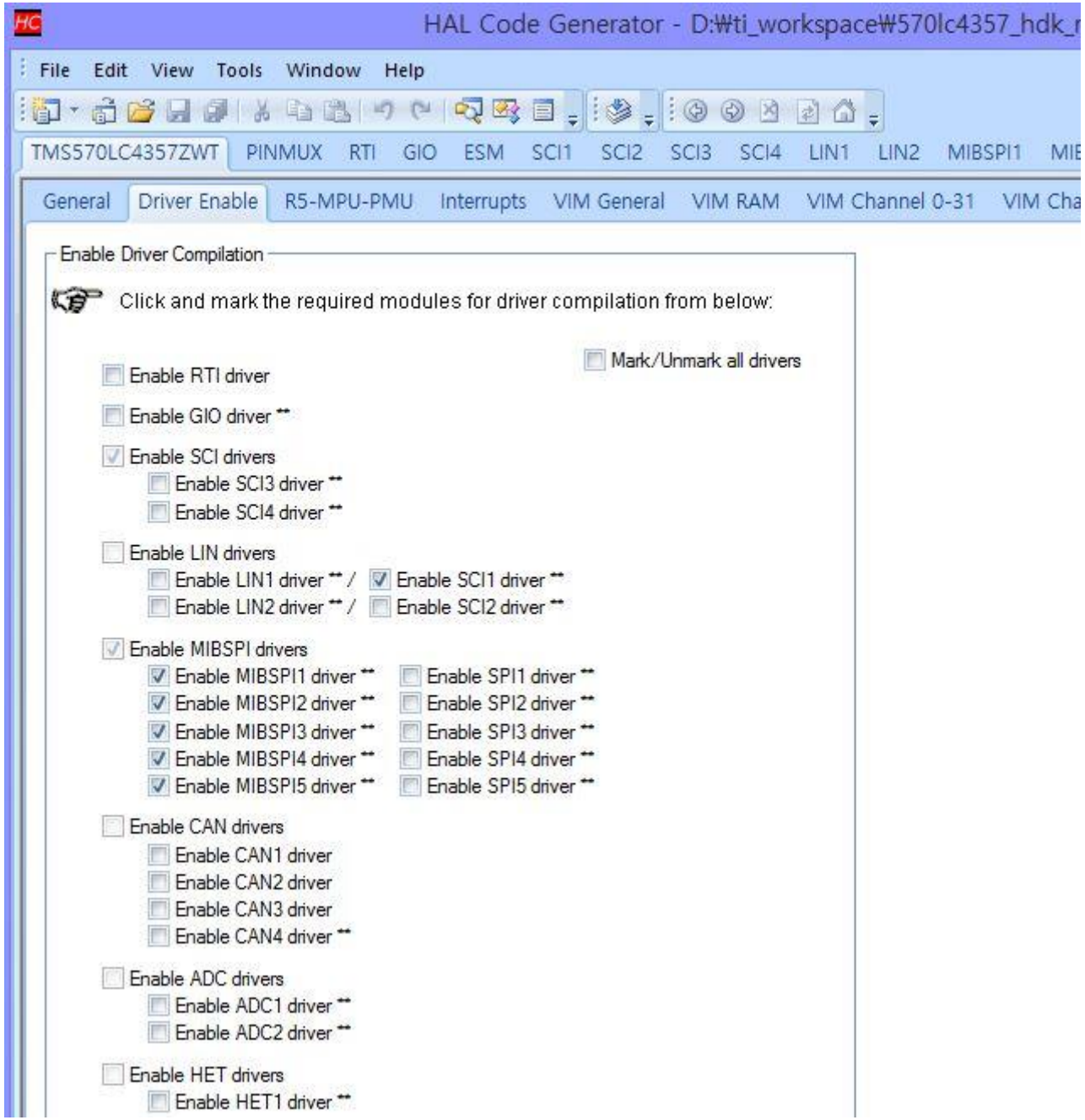
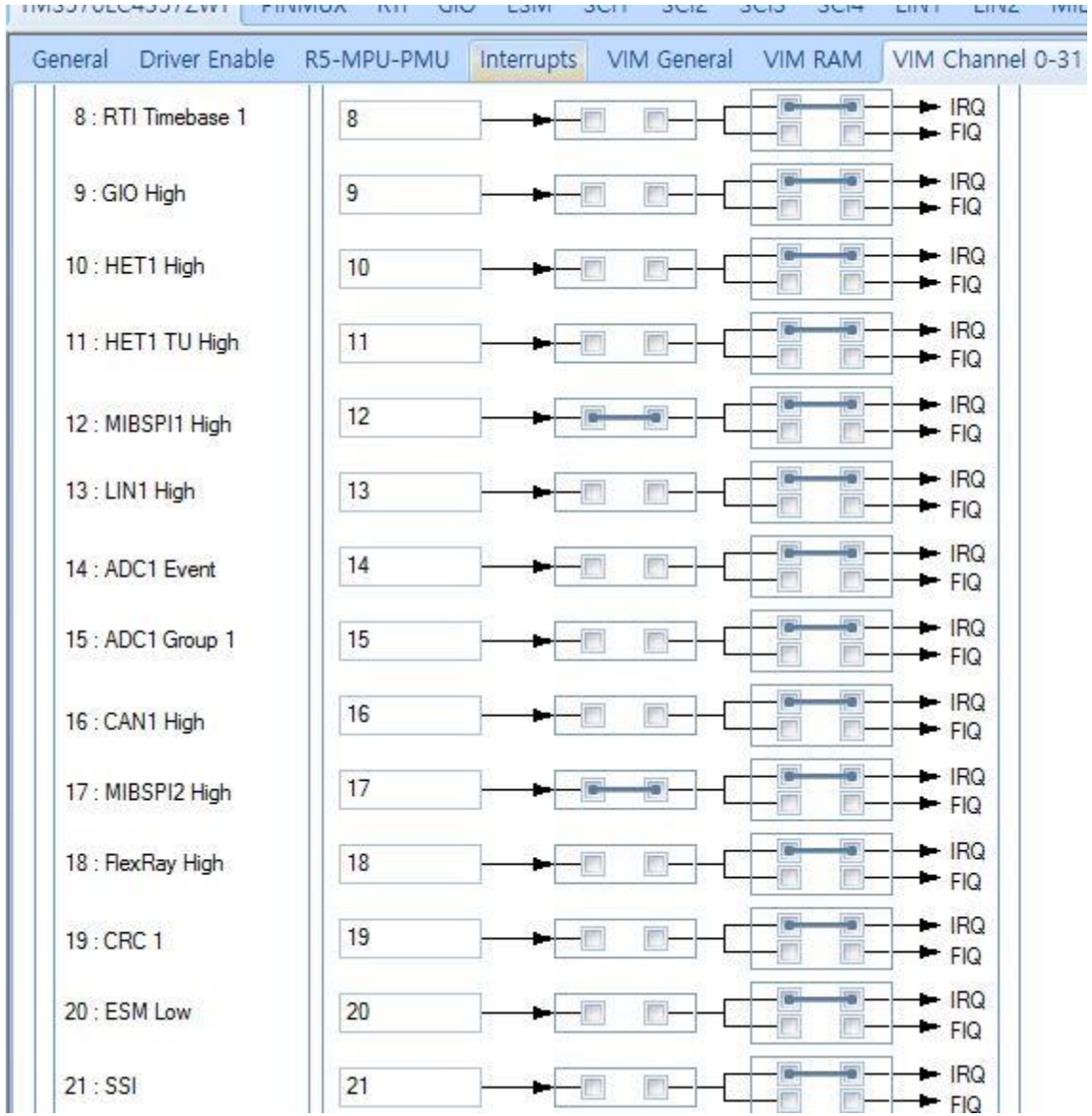
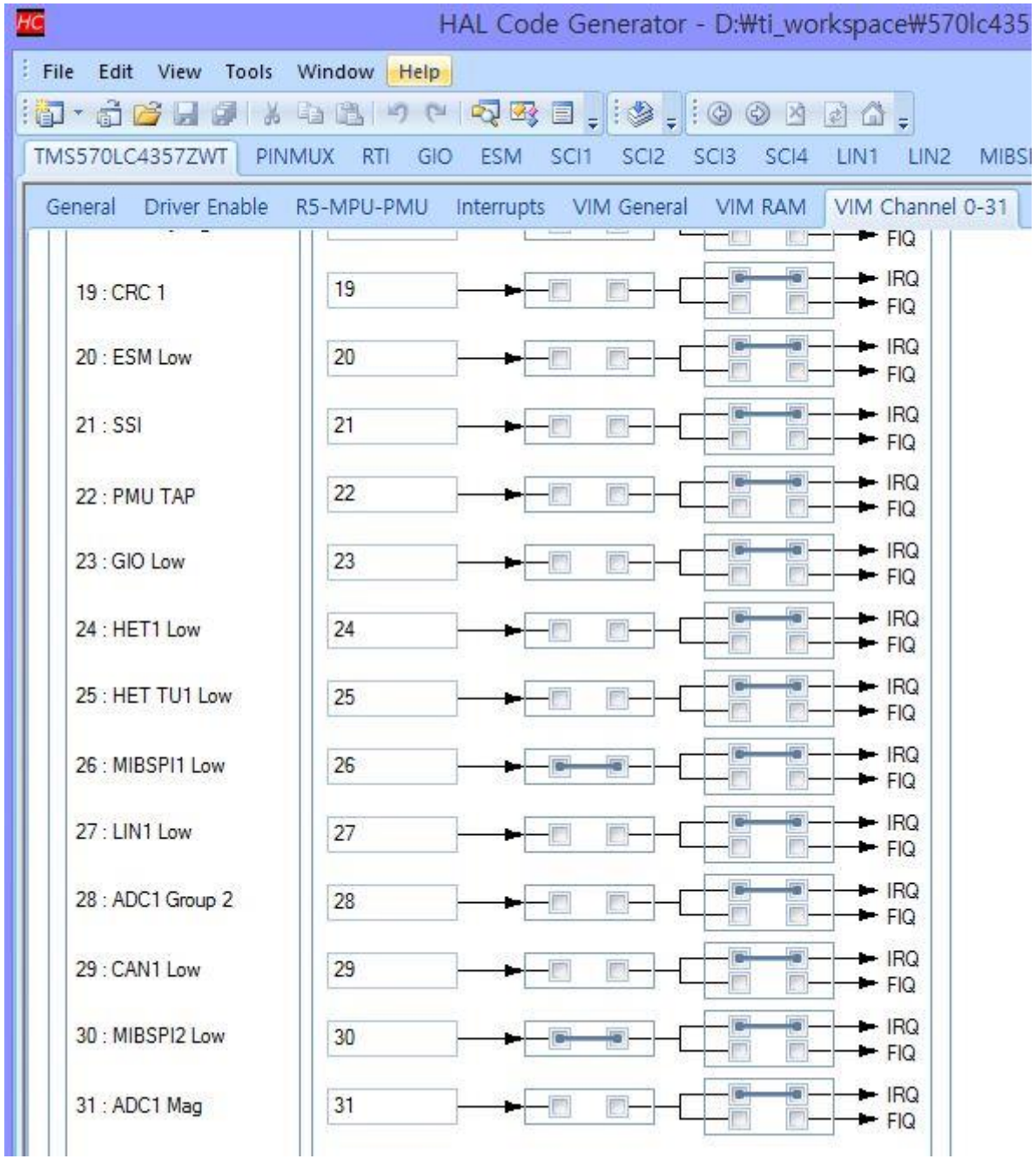
Xilinx Zynq FPGA, TI DSP, MCU 프로그래밍 및 회로 설계 전문가 과정

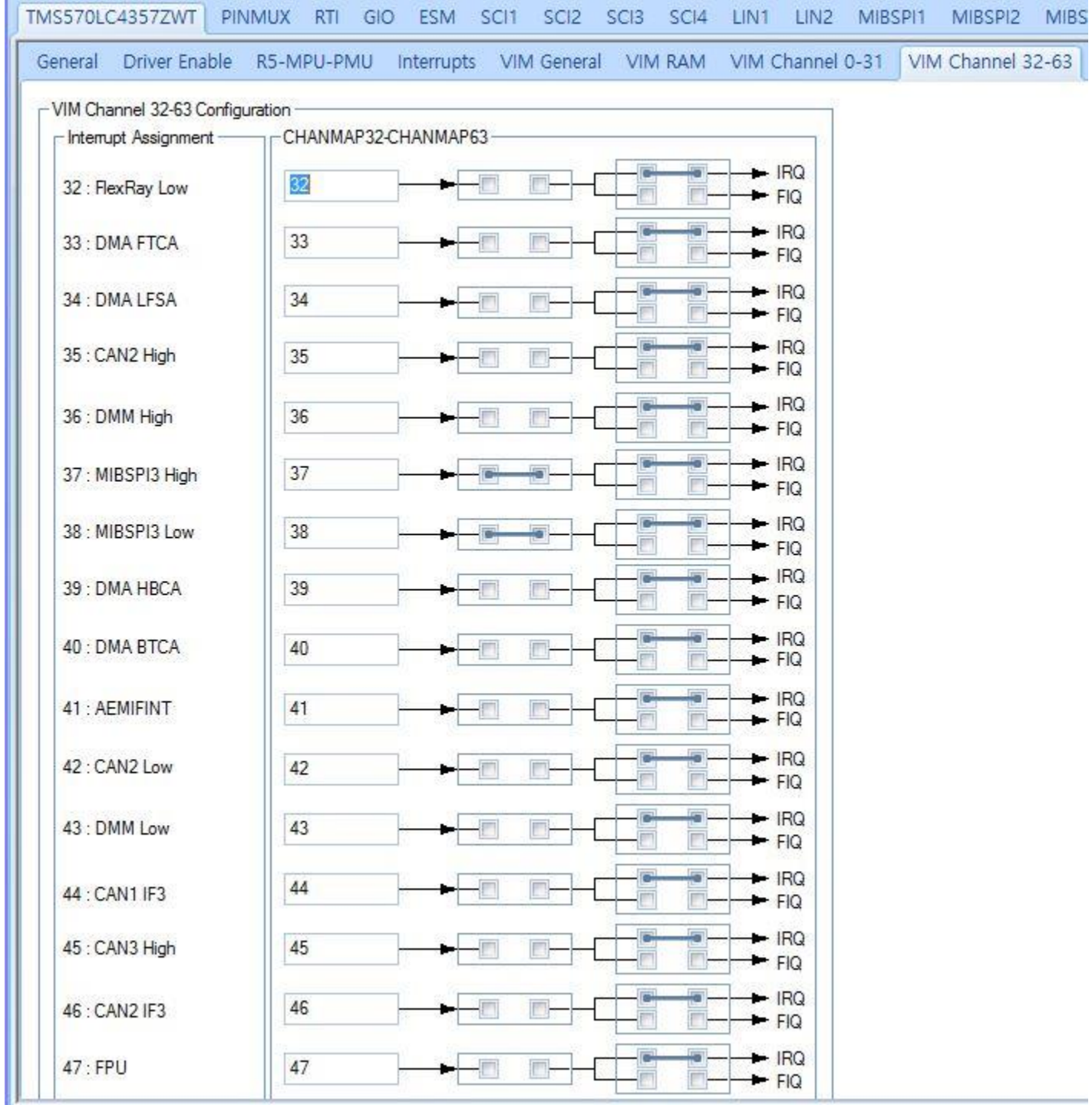
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

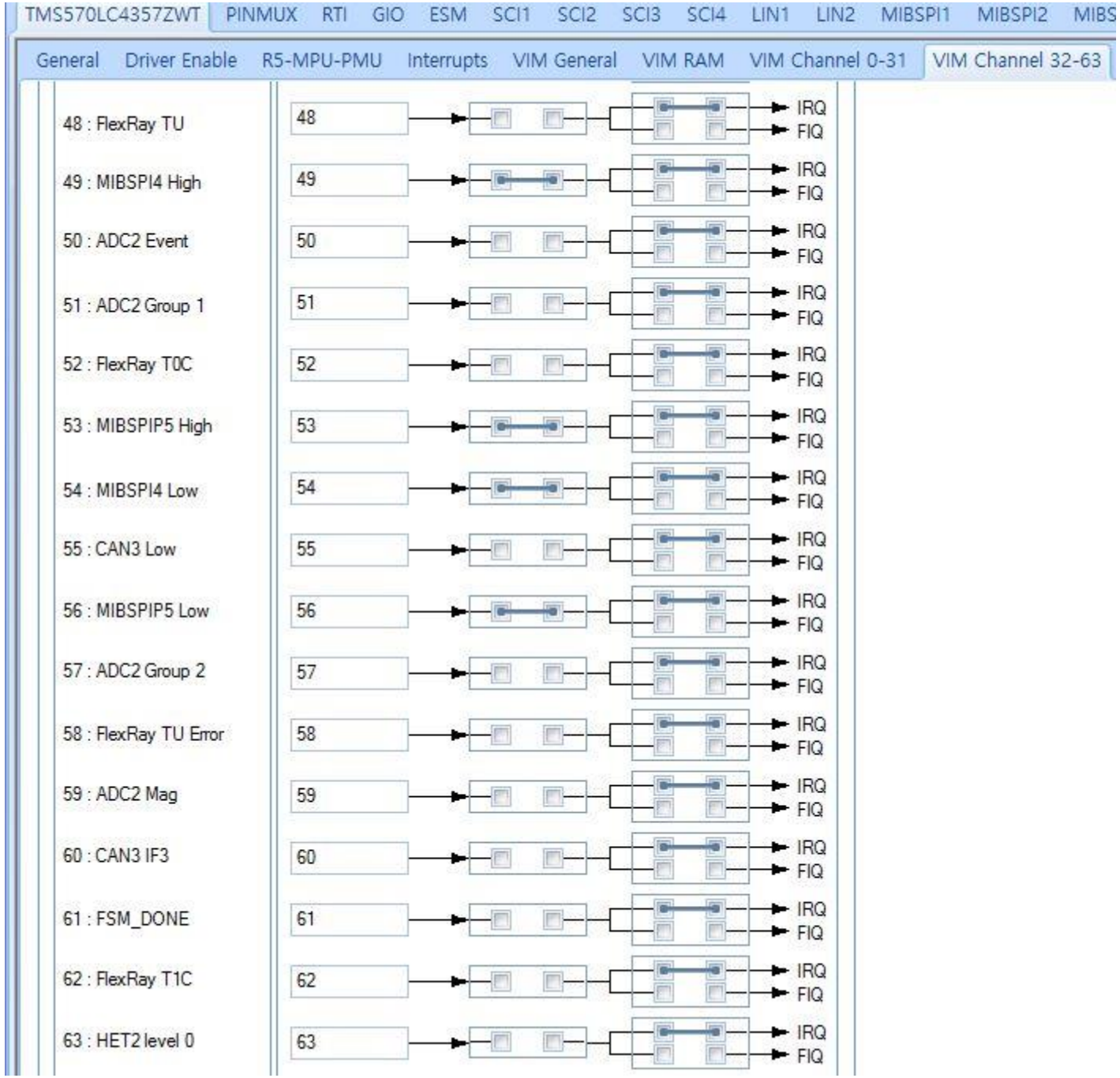
SPI Loopback Test

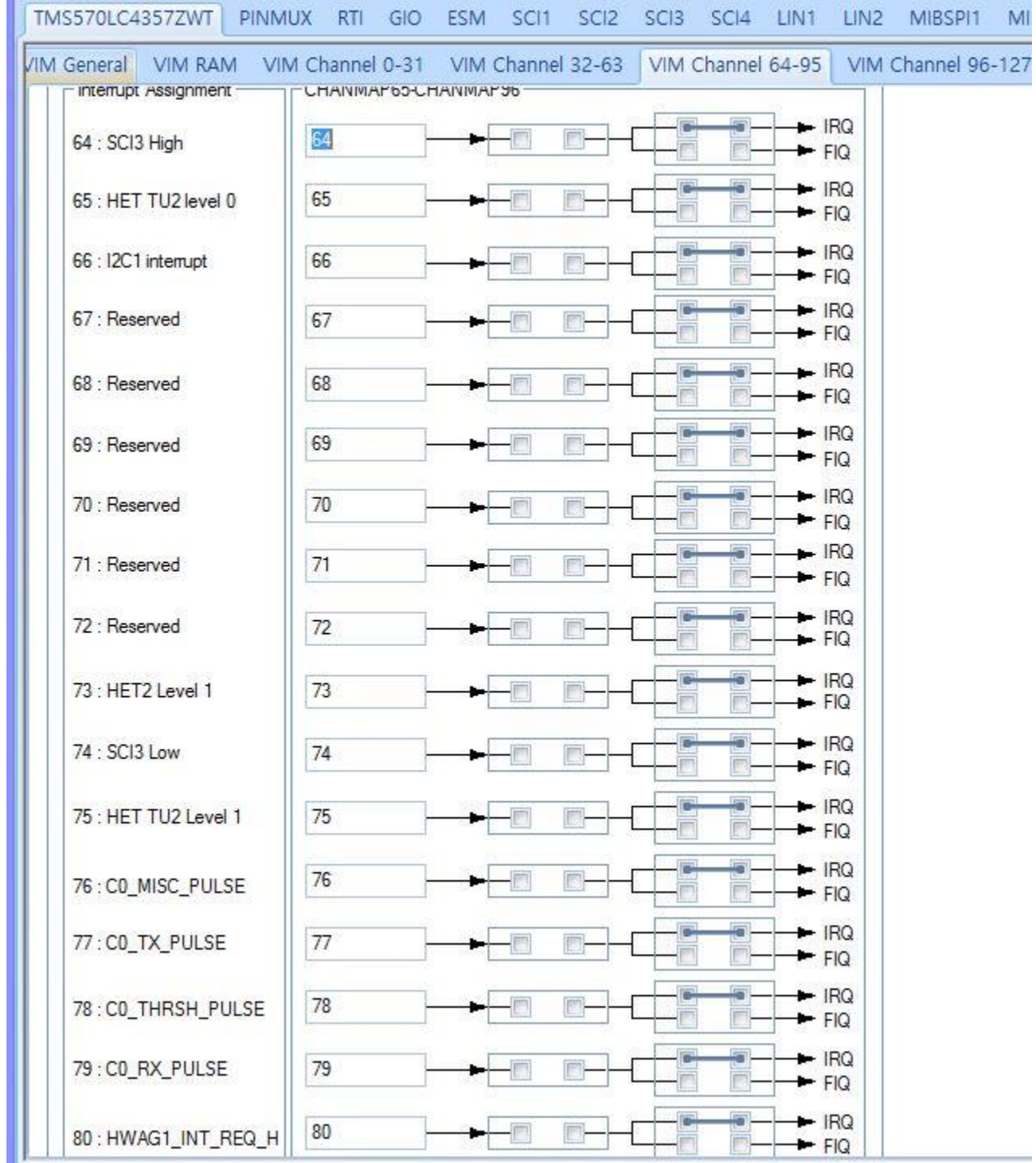

HAL Code Generator - D:¥ti_workspace¥570lc4357_hdk_r
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIE
General Driver Enable R5-MPU-PMU Interrupts VIM General VIM RAM VIM Channel 0-31 VIM Cha
Enable Driver Compilation
Click and mark the required modules for driver compilation from below:
Mark/Unmark all drivers
Enable RTI driver
Enable GIO driver **
Enable SCI drivers
Enable SCI3 driver **
Enable SCI4 driver **
Enable LIN drivers
Enable LIN1 driver ** / Enable SCI1 driver **
Enable LIN2 driver ** / Enable SCI2 driver **
Enable MIBSPI drivers
Enable MIBSPI1 driver ** Enable SPI1 driver **
Enable MIBSPI2 driver ** Enable SPI2 driver **
Enable MIBSPI3 driver ** Enable SPI3 driver **
Enable MIBSPI4 driver ** Enable SPI4 driver **
Enable MIBSPI5 driver ** Enable SPI5 driver **
Enable CAN drivers
Enable CAN1 driver
Enable CAN2 driver
Enable CAN3 driver
Enable CAN4 driver **
Enable ADC drivers
Enable ADC1 driver **
Enable ADC2 driver **
Enable HET drivers
Enable HET1 driver **

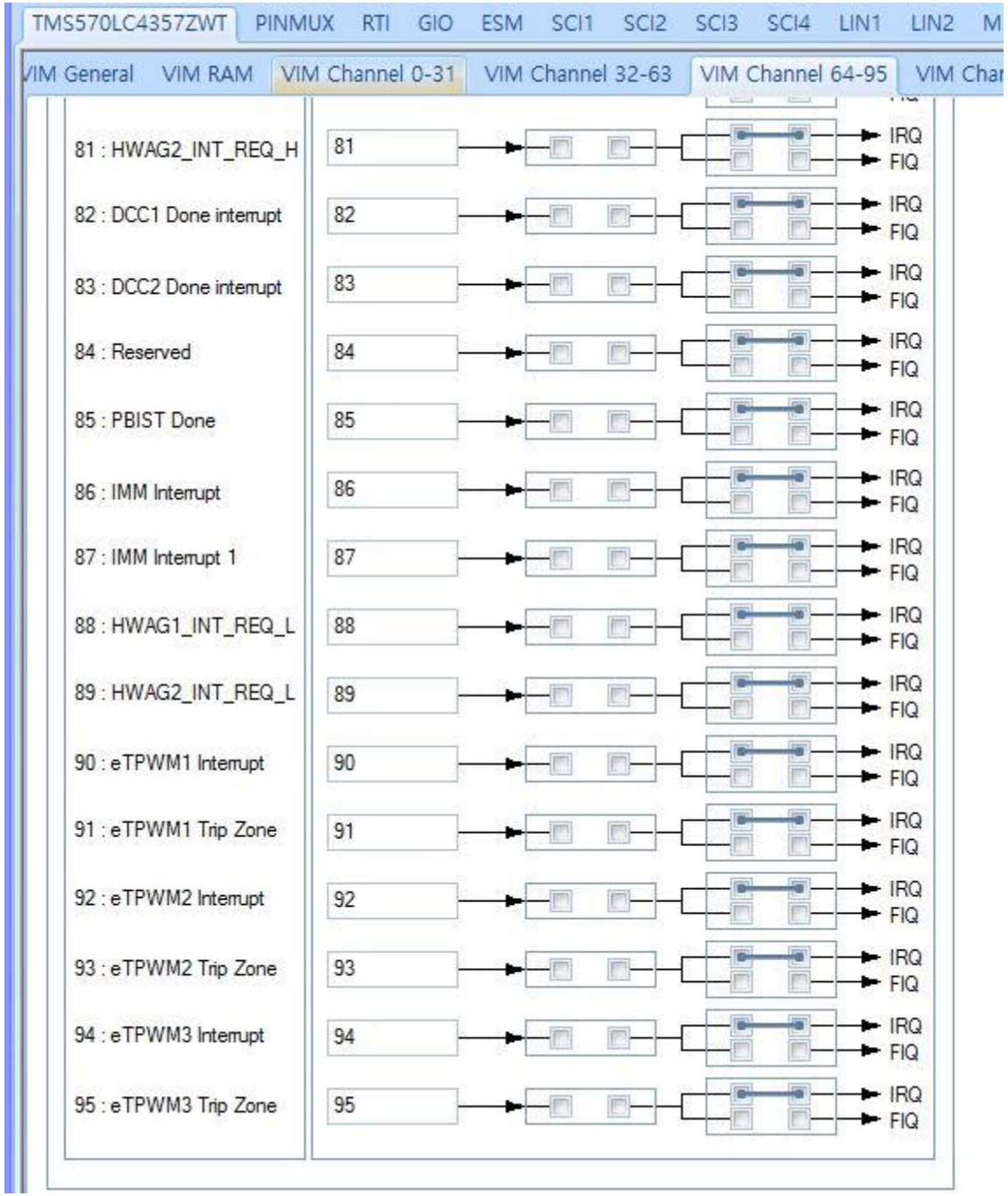

General
Driver Enable
R5-MPU-PMU
Interrupts
VIM General
VIM RAM
VIM Channel 0-31
8 : RTI Timebase 1
9 : GIO High
10 : HET1 High
11 : HET1 TU High
12 : MIBSPI1 High
13 : LIN1 High
14 : ADC1 Event
15 : ADC1 Group 1
16 : CAN1 High
17 : MIBSPI2 High
18 : FlexRay High
19 : CRC 1
20 : ESM Low
21 : SSI
IRQ
FIQ

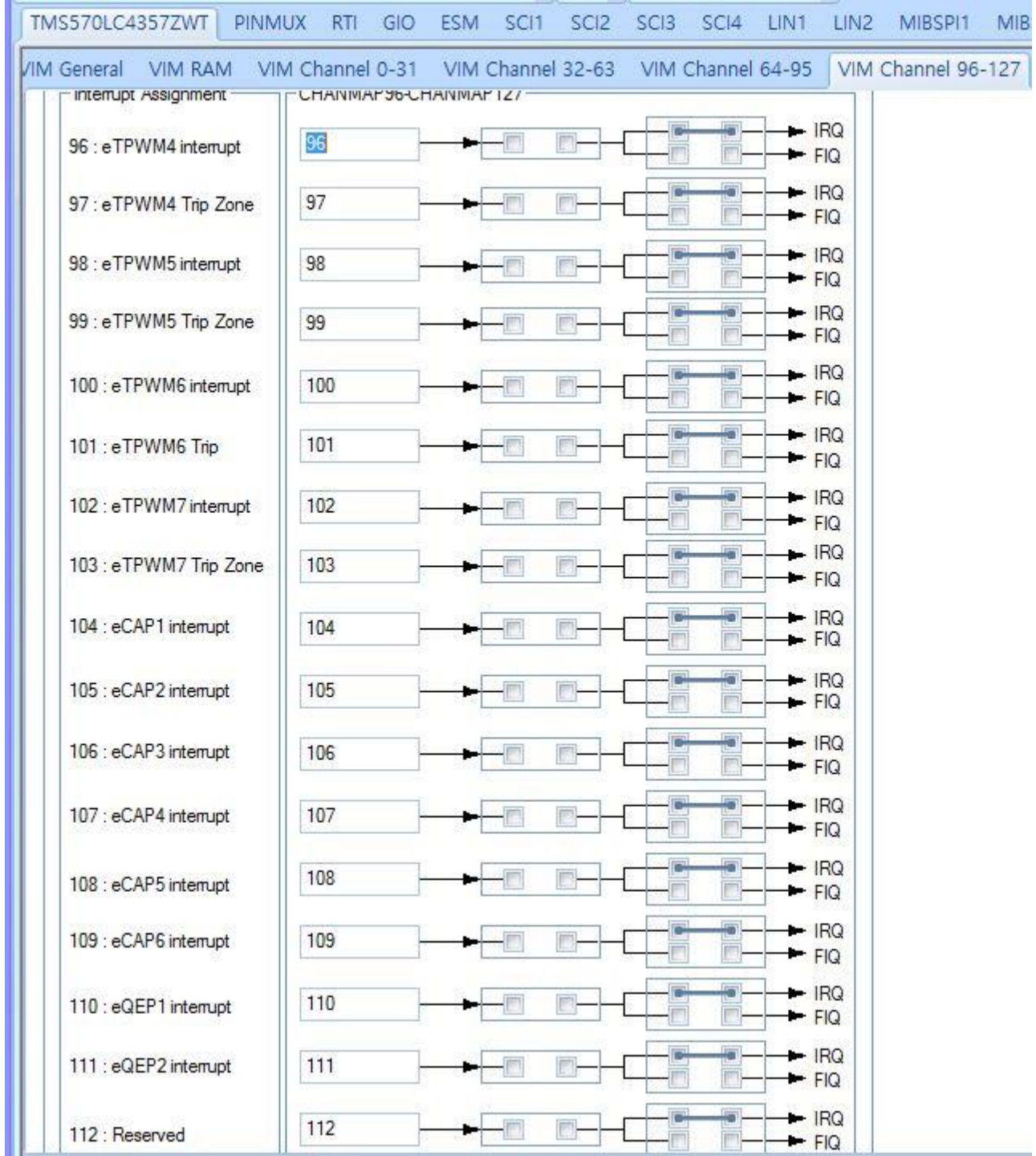

HAL Code Generator - D:₩ti_workspace₩570lc435
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSI
General Driver Enable R5-MPU-PMU Interrupts VIM General VIM RAM VIM Channel 0-31
FIQ
19 : CRC 1
19
IRQ
FIQ
20 : ESM Low
20
IRQ
FIQ
21 : SSI
21
IRQ
FIQ
22 : PMU TAP
22
IRQ
FIQ
23 : GIO Low
23
IRQ
FIQ
24 : HET1 Low
24
IRQ
FIQ
25 : HET TU1 Low
25
IRQ
FIQ
26 : MIBSPI1 Low
26
IRQ
FIQ
27 : LIN1 Low
27
IRQ
FIQ
28 : ADC1 Group 2
28
IRQ
FIQ
29 : CAN1 Low
29
IRQ
FIQ
30 : MIBSPI2 Low
30
IRQ
FIQ
31 : ADC1 Mag
31
IRQ
FIQ

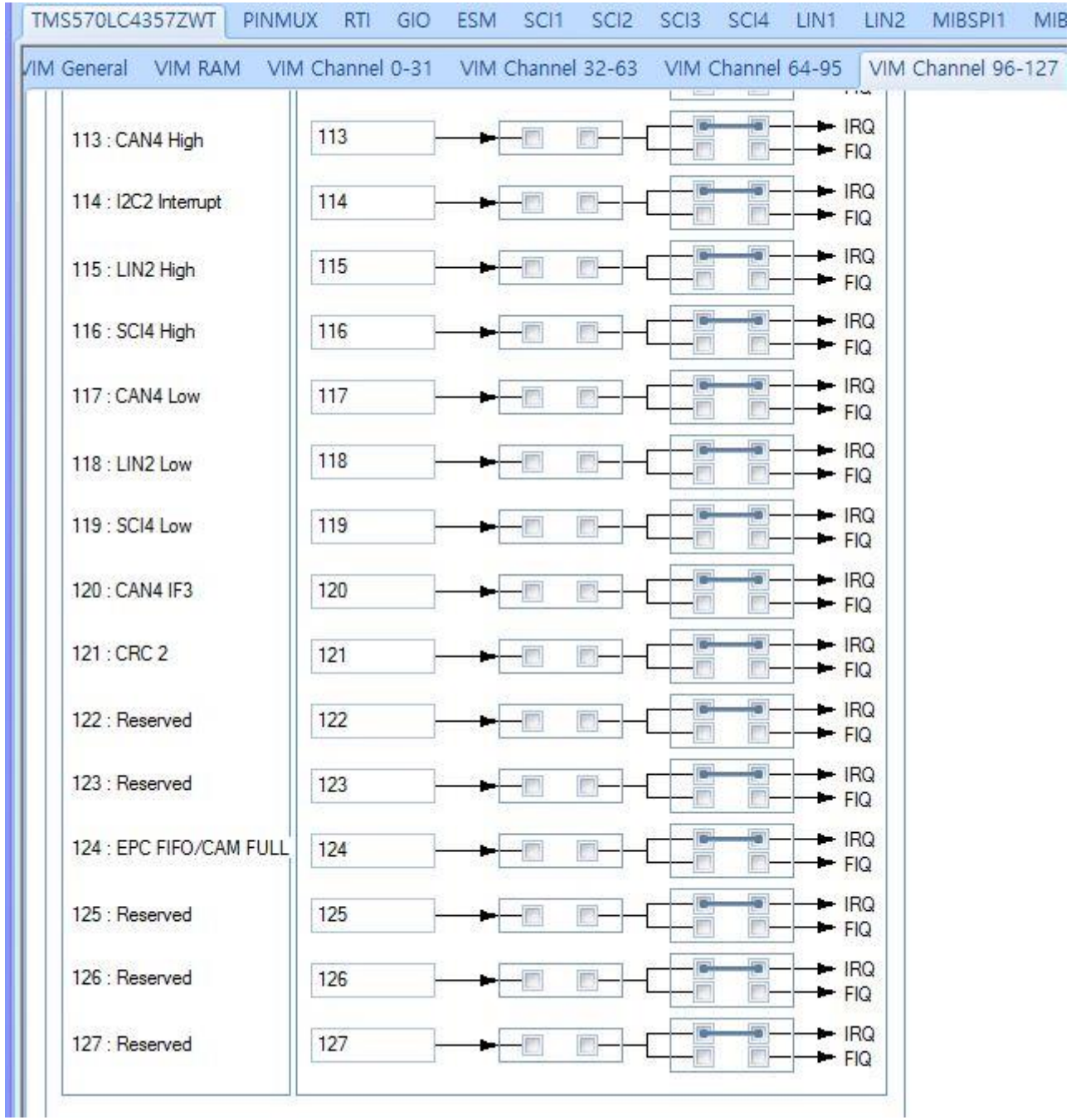

TMS570LC4357ZWT
PINMUX
RTI
GIO
ESM
SCI1
SCI2
SCI3
SCI4
LIN1
LIN2
MIBSPI1
MIBSPI2
MIBS
General
Driver Enable
R5-MPU-PMU
Interrupts
VIM General
VIM RAM
VIM Channel 0-31
VIM Channel 32-63
VIM Channel 32-63 Configuration
Interrupt Assignment
CHANMAP32-CHANMAP63
32 : FlexRay Low
33 : DMA FTCA
34 : DMA LFSA
35 : CAN2 High
36 : DMM High
37 : MIBSPI3 High
38 : MIBSPI3 Low
39 : DMA HBCA
40 : DMA BTCA
41 : AEMIFINT
42 : CAN2 Low
43 : DMM Low
44 : CAN1 IF3
45 : CAN3 High
46 : CAN2 IF3
47 : FPU
IRQ
FIQ

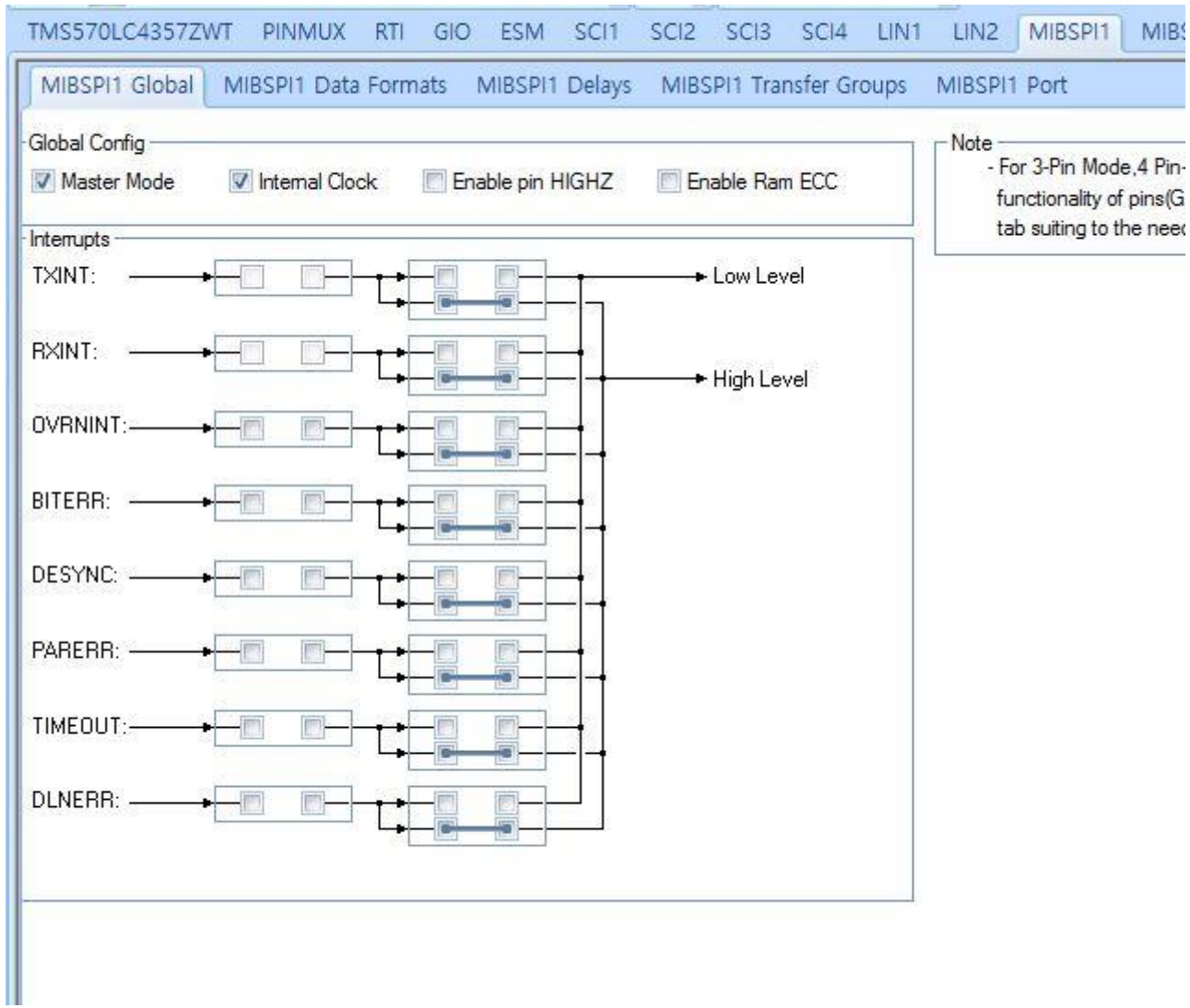

TMS570LC4357ZWT
PINMUX
RTI
GIO
ESM
SCI1
SCI2
SCI3
SCI4
LIN1
LIN2
MIBSPI1
MIBSPI2
MIBS
General
Driver Enable
R5-MPU-PMU
Interrupts
VIM General
VIM RAM
VIM Channel 0-31
VIM Channel 32-63
48 : FlexRay TU
49 : MIBSPI4 High
50 : ADC2 Event
51 : ADC2 Group 1
52 : FlexRay T0C
53 : MIBSPIP5 High
54 : MIBSPI4 Low
55 : CAN3 Low
56 : MIBSPIP5 Low
57 : ADC2 Group 2
58 : FlexRay TU Error
59 : ADC2 Mag
60 : CAN3 IF3
61 : FSM_DONE
62 : FlexRay T1C
63 : HET2 level 0
IRQ
FIQ


TMS570LC4357ZWT
PINMUX
RTI
GIO
ESM
SCI1
SCI2
SCI3
SCI4
LIN1
LIN2
MIBSPI1
MI
VIM General
VIM RAM
VIM Channel 0-31
VIM Channel 32-63
VIM Channel 64-95
VIM Channel 96-127
Interrupt Assignment
CHANMAP65-CHANMAP96
64 : SCI3 High
65 : HET TU2 level 0
66 : I2C1 interrupt
67 : Reserved
68 : Reserved
69 : Reserved
70 : Reserved
71 : Reserved
72 : Reserved
73 : HET2 Level 1
74 : SCI3 Low
75 : HET TU2 Level 1
76 : C0_MISC_PULSE
77 : C0_TX_PULSE
78 : C0_THRSH_PULSE
79 : C0_RX_PULSE
80 : HWAG1_INT_REQ_H
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
IRQ
FIQ


TMS570LC4357ZWT
PINMUX
RTI
GIO
ESM
SCI1
SCI2
SCI3
SCI4
LIN1
LIN2
M
VIM General
VIM RAM
VIM Channel 0-31
VIM Channel 32-63
VIM Channel 64-95
81 : HWAG2_INT_REQ_H
82 : DCC1 Done interrupt
83 : DCC2 Done interrupt
84 : Reserved
85 : PBIST Done
86 : IMM Interrupt
87 : IMM Interrupt 1
88 : HWAG1_INT_REQ_L
89 : HWAG2_INT_REQ_L
90 : eTPWM1 Interrupt
91 : eTPWM1 Trip Zone
92 : eTPWM2 Interrupt
93 : eTPWM2 Trip Zone
94 : eTPWM3 Interrupt
95 : eTPWM3 Trip Zone
IRQ
FIQ


TMS570LC4357ZWT
PINMUX
RTI
GIO
ESM
SCI1
SCI2
SCI3
SCI4
LIN1
LIN2
MIBSPI1
MIB
VIM General
VIM RAM
VIM Channel 0-31
VIM Channel 32-63
VIM Channel 64-95
VIM Channel 96-127
Interrupt Assignment
CHANMAP96-CHANMAP127
96 : eTPWM4 interrupt
97 : eTPWM4 Trip Zone
98 : eTPWM5 interrupt
99 : eTPWM5 Trip Zone
100 : eTPWM6 interrupt
101 : eTPWM6 Trip
102 : eTPWM7 interrupt
103 : eTPWM7 Trip Zone
104 : eCAP1 interrupt
105 : eCAP2 interrupt
106 : eCAP3 interrupt
107 : eCAP4 interrupt
108 : eCAP5 interrupt
109 : eCAP6 interrupt
110 : eQEP1 interrupt
111 : eQEP2 interrupt
112 : Reserved
IRQ
FIQ


TMS570LC4357ZWT
PINMUX
RTI
GIO
ESM
SCI1
SCI2
SCI3
SCI4
LIN1
LIN2
MIBSPI1
MIB
VIM General
VIM RAM
VIM Channel 0-31
VIM Channel 32-63
VIM Channel 64-95
VIM Channel 96-127
113 : CAN4 High
113
IRQ
FIQ
114 : I2C2 Interrupt
114
IRQ
FIQ
115 : LIN2 High
115
IRQ
FIQ
116 : SCI4 High
116
IRQ
FIQ
117 : CAN4 Low
117
IRQ
FIQ
118 : LIN2 Low
118
IRQ
FIQ
119 : SCI4 Low
119
IRQ
FIQ
120 : CAN4 IF3
120
IRQ
FIQ
121 : CRC 2
121
IRQ
FIQ
122 : Reserved
122
IRQ
FIQ
123 : Reserved
123
IRQ
FIQ
124 : EPC FIFO/CAM FULL
124
IRQ
FIQ
125 : Reserved
125
IRQ
FIQ
126 : Reserved
126
IRQ
FIQ
127 : Reserved
127
IRQ
FIQ

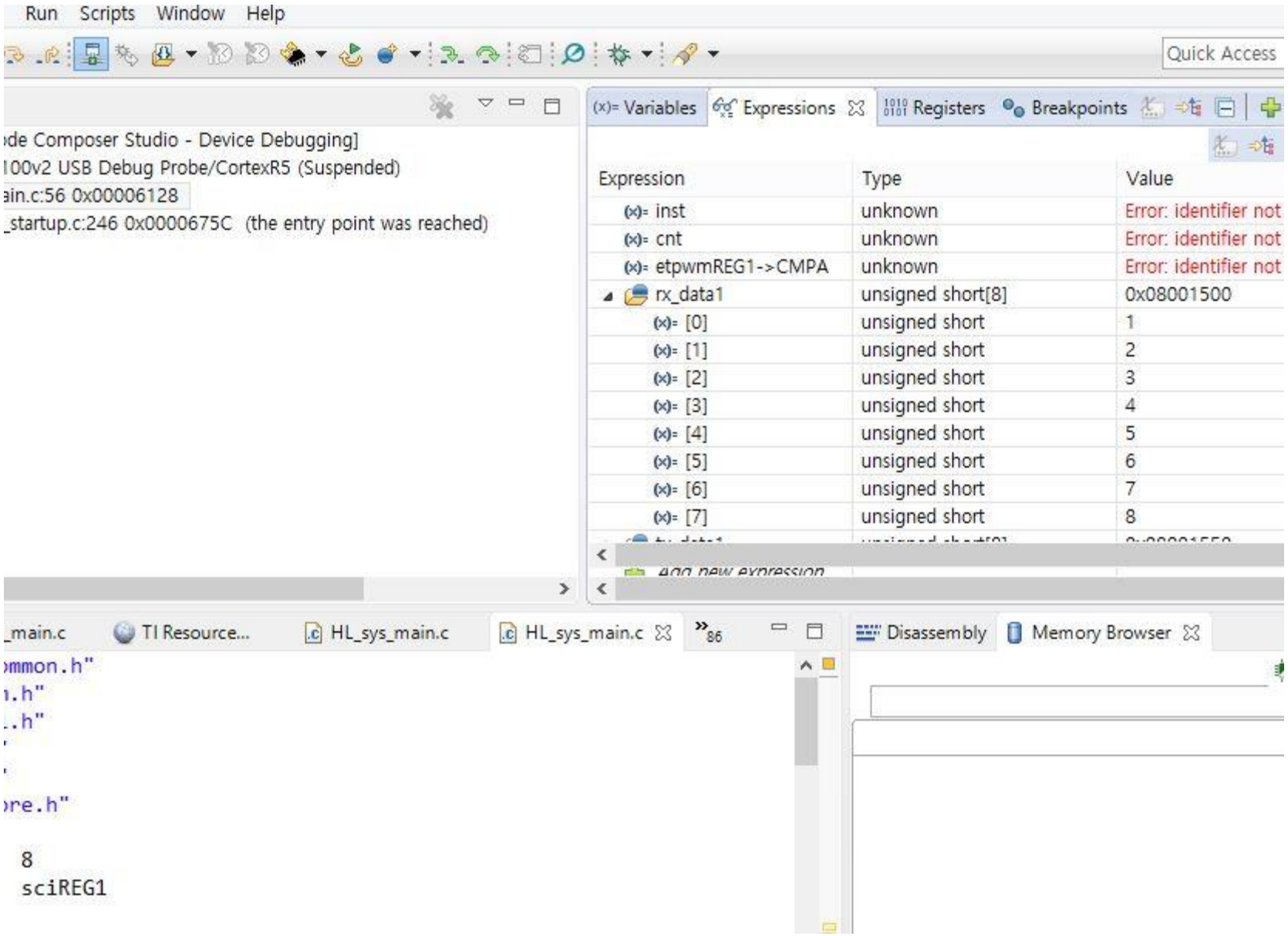

TMS570LC4357ZWT
PINMUX
RTI
GIO
ESM
SCI1
SCI2
SCI3
SCI4
LIN1
LIN2
MIBSPI1
MIBS
MIBSPI1 Global
MIBSPI1 Data Formats
MIBSPI1 Delays
MIBSPI1 Transfer Groups
MIBSPI1 Port
Global Config
Master Mode
Internal Clock
Enable pin HIGHZ
Enable Ram ECC
Note
- For 3-Pin Mode,4 Pin-
functionality of pins(G
tab suiting to the nee
Interrupts
TXINT:
RXINT:
OVRNINT:
BITERR:
DESYNC:
PARERR:
TIMEOUT:
DLNERR:
Low Level
High Level

```c
#include "HL_sys_common.h"
#include "HL_system.h"
#include "HL_mibspi.h"
#include "HL_esm.h"
#include "HL_sci.h"
#include "HL_sys_core.h"

#define D_COUNT     8
#define UART        sciREG1

uint32 cnt=0, error =0, tx_done =0;
uint16 tx_data1[D_COUNT] = {1,2,3,4,5,6,7,8};
uint16 tx_data2[D_COUNT] = {11,12,13,14,15,16,17,18};
uint16 tx_data3[D_COUNT] = {21,22,23,24,25,26,27,28};
uint16 tx_data4[D_COUNT] = {31,32,33,34,35,36,37,38};
uint16 tx_data5[D_COUNT] = {41,42,43,44,45,46,47,48};
uint16 rx_data1[D_COUNT] = {0};
uint16 rx_data2[D_COUNT] = {0};
uint16 rx_data3[D_COUNT] = {0};
uint16 rx_data4[D_COUNT] = {0};
uint16 rx_data5[D_COUNT] = {0};

void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 length)
{
    while(length--)
    {
        while((UART->FLR & 0x4) == 4)
            ;
        sciSendByte(UART, *text++);
    }
}

void wait(uint32 time)
{
    time--;
}
```

R5F_UART/source/HL_sys_main.c

```c
void main(void)
{
    _enable_IRQ_interrupt_();

    sciInit();
    mibspiInit();

    mibspiEnableLoopback(mibspiREG1, Digital_Lbk);
    mibspiEnableLoopback(mibspiREG2, Digital_Lbk);
    mibspiEnableLoopback(mibspiREG3, Digital_Lbk);
    mibspiEnableLoopback(mibspiREG4, Digital_Lbk);
    mibspiEnableLoopback(mibspiREG5, Digital_Lbk);

    mibspiEnableGroupNotification(mibspiREG1, 0, 1);
    mibspiEnableGroupNotification(mibspiREG2, 0, 1);
    mibspiEnableGroupNotification(mibspiREG3, 0, 1);
    mibspiEnableGroupNotification(mibspiREG4, 0, 1);
    mibspiEnableGroupNotification(mibspiREG5, 0, 1);

    mibspiSetData(mibspiREG1, 0, &tx_data1[0]);
    mibspiSetData(mibspiREG2, 0, &tx_data2[0]);
    mibspiSetData(mibspiREG3, 0, &tx_data3[0]);
    mibspiSetData(mibspiREG4, 0, &tx_data4[0]);
    mibspiSetData(mibspiREG5, 0, &tx_data5[0]);

    mibspiTransfer(mibspiREG1, 0);
    mibspiTransfer(mibspiREG2, 0);
    mibspiTransfer(mibspiREG3, 0);
    mibspiTransfer(mibspiREG4, 0);
    mibspiTransfer(mibspiREG5, 0);

    //sciDisplayText(UART, &rx_data1[0], D_COUNT);

    while(1)
        ;
}
```

```
75 void mibspiGroupNotification(mibspiBASE_t *mibspi, uint32 group)
76 {
77     uint16 * data;
78     if(mibspi==mibspiREG1)
79     {
80         data = &rx_data1[0];
81     }
82     if(mibspi==mibspiREG2)
83     {
84         data = &rx_data2[0];
85     }
86     if(mibspi==mibspiREG3)
87     {
88         data = &rx_data3[0];
89     }
90     if(mibspi==mibspiREG4)
91     {
92         data = &rx_data4[0];
93     }
94     if(mibspi==mibspiREG5)
95     {
96         data = &rx_data5[0];
97     }
98
99     mibspiGetData(mibspi, group, data);
100 }
101
```


Run Scripts Window Help
Quick Access
de Composer Studio - Device Debugging]
100v2 USB Debug Probe/CortexR5 (Suspended)
ain.c:56 0x00006128
_startup.c:246 0x0000675C (the entry point was reached)
Variables
Expressions
Registers
Breakpoints
Expression
Type
Value
inst
unknown
Error: identifier not
cnt
unknown
Error: identifier not
etpwmREG1->CMPA
unknown
Error: identifier not
rx_data1
unsigned short[8]
0x08001500
[0]
unsigned short
1
[1]
unsigned short
2
[2]
unsigned short
3
[3]
unsigned short
4
[4]
unsigned short
5
[5]
unsigned short
6
[6]
unsigned short
7
[7]
unsigned short
8
Add new expression
_main.c
TI Resource...
HL_sys_main.c
HL_sys_main.c
Disassembly
Memory Browser
ommon.h"
1.h"
..h"
ore.h"
8
sciREG1