

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

강사 – Innova Lee(이상훈)

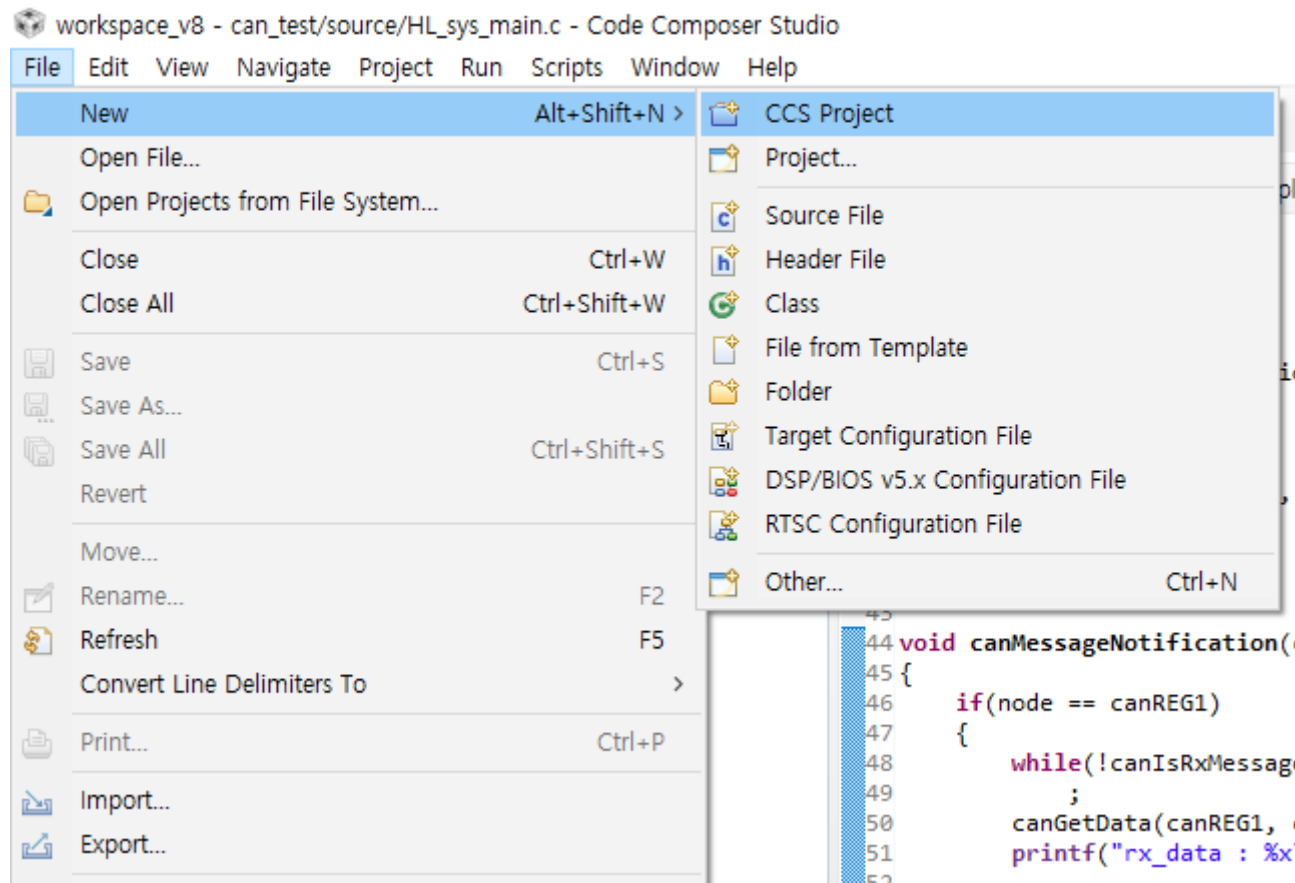
gcccompil3r@gmail.com

학생 – Hyungjoo Kim(김형주)

mihaelkel@naver.com

UART 통신 (DSP ↔ MCU)

- 먼저 MCU 의 UART 프로그램을 작성한다 .



CCS Project

Create a new CCS Project.



Target: <select or type filter text> TMS570LC43xx

Connection: Texas Instruments XDS100v2 USB Debug Probe Verify...

Cortex R [ARM]

Project name: uart_test

☒ Use default location

Location: C:\Users\Howard\workspace_v8\uart_test Browse...

Compiler version: TI v18.1.2.LTS More...

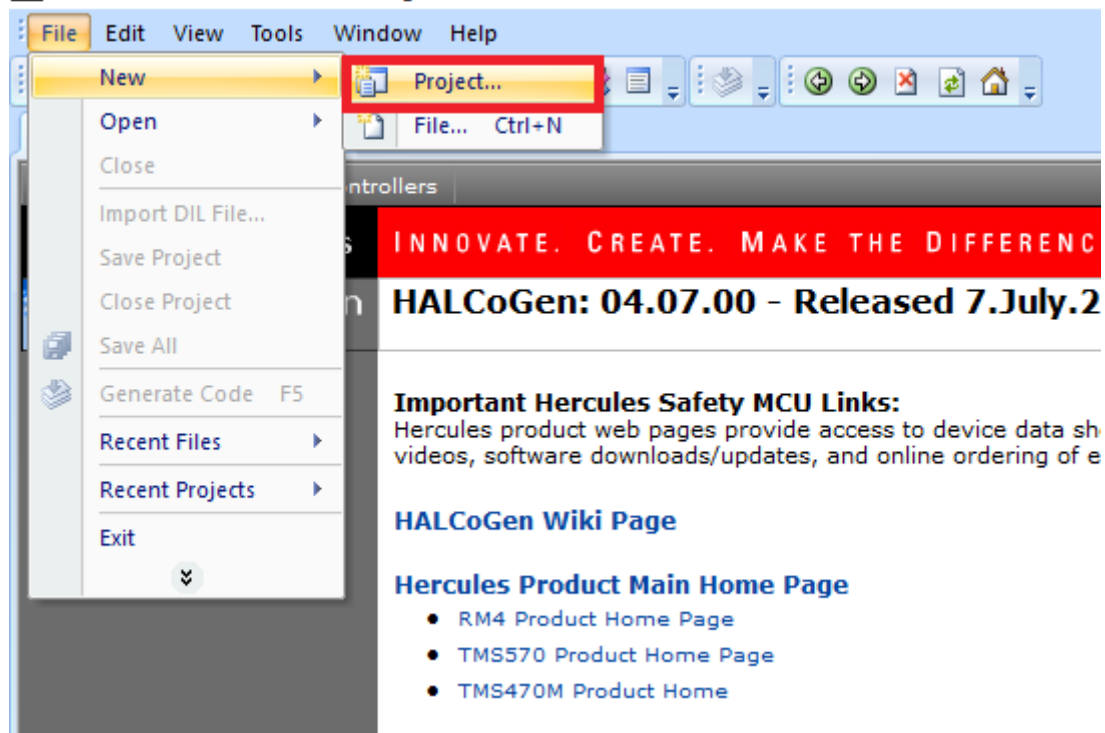
Tool-chain

Project templates and examples

type filter text

- Empty Projects
 - Empty Project
 - Empty Project (with main.c)
 - Empty Assembly-only Project
 - Empty RTSC Project
- Basic Examples
 - Hello World

Creates an empty project initialized for the selected device.



New Project

Family:

- TMS570LS04x
- TMS570LS03x
- TMS570LS02x
- RM42x
- RM41x
- TMS570LS09x_07x
- RM44x
- TMS570LC43x**
- RM57Lx

Device:

- TMS570LC4357ZWT**
- TMS570LC4357ZWT_FREERTOS

Name: **uart_test**

Location: **C:\Users\Howard\workspace_v8\uart_test**

☐ Create directory for project

Project will be created at: C:\Users\Howard\workspace_v8\uart_test.

Tools: Texas Instruments Tools

OK Cancel

Enable Driver Compilation



Click and mark the required modules for driver compilation from below:

☐ Enable RTI driver

☐ Mark/Unmark all drivers

☐ 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 **

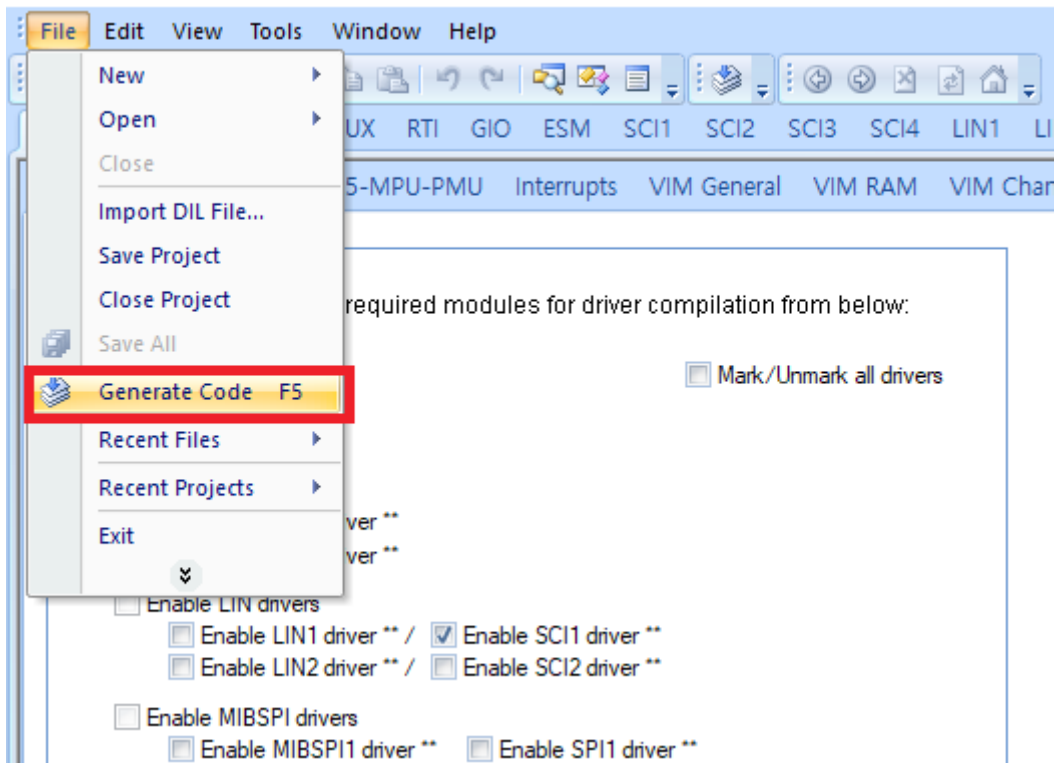
☐ Enable CAN drivers

☐ Enable CAN1 driver

☐ Enable CAN2 driver

☐ Enable CAN3 driver

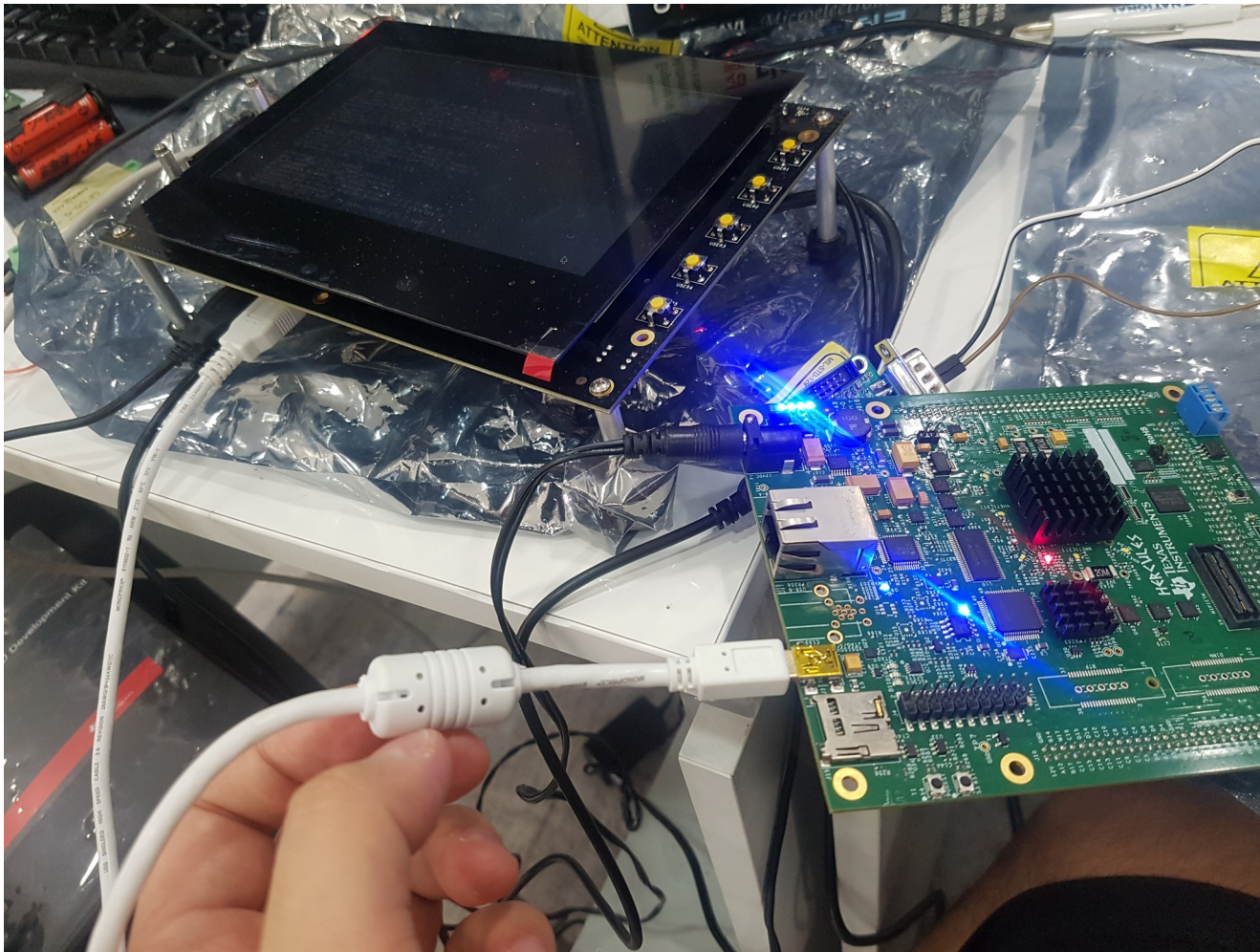
☐ Enable CAN4 driver **



- 아래와 같이 코드를 작성한 후 flash 를 한다 .

```
Getting Started  Resource Explorer  HL_sys_main.c  *HL_sys_main.c  ⌕
1 #include "HL_sys_common.h"
2 #include "HL_sci.h"
3
4 int main(void)
5 {
6     uint8 msg[8] = {'A', 'B', 'C', '\r', '\n', '\0'};
7     sciInit();
8
9     while(1)
10    {
11        while(!sciIsTxReady(sciREG3))
12            ;
13        msg[0] = (uint8)sciReceiveByte(sciREG3);
14        /*
15        send_msg(sciREG1, msg, 8);
16    }
17
18    return 0;
19 }
20 void send_msg(sciBASE_t* sci, uint8* msg, uint8 length)
21 {
22     int i;
23     for(i = 0; i < length; i++)
24     {
25         while(!sciIsTxReady(sci))
26             ;
27         sciSendByte(sci, msg[i]);
28     }
29 }
```

- 아래와 같이 MCU 와 DSP 를 연결한다 .



- ccs 의 Remote Systems 영역으로 들어가서 , 아래와 같이 dsp 프로그램을 작성한다 .

```
1#include <stdio.h>
2#include <stdlib.h>
3#include <string.h>
4#include <unistd.h>
5#include <sys/types.h>
6#include <sys/poll.h>
7#include <termios.h>           // B115200, CS8 등 상수 정의
8#include <fcntl.h>             // O_RDWR , O_NOCTTY 등의 상수 정의
9|
10int main( void)
11{
12    int    fd;
13    int    ndx;
14    int    cnt;
15    char   buf[1024];
16    struct termios  newtio;
17    struct pollfd   poll_events;    // 체크할 event 정보를 갖는 struct
18    int    poll_state;
19
20    // 시리얼 포트를 open
21
22    fd = open( "/dev/ttyUSB1", O_RDWR | O_NOCTTY | O_NONBLOCK );    // 디바이스를 open 한다.
23    if ( 0 > fd)
24    {
25        printf("open error\n");
26        return -1;
27    }
28
29    // 시리얼 포트 통신 환경 설정
30
31    memset( &newtio, 0, sizeof(newtio) );
32    newtio.c_cflag      = B9600 | CS8 | CLOCAL | CREAD;
33    newtio.c_oflag      = 0;
34    newtio.c_lflag      = 0;
35    newtio.c_cc[VTIME]  = 0;
36    newtio.c_cc[VMIN]   = 1;
```

```

43 // poll 사용을 위한 준비
44
45 poll_events.fd      = fd;
46 poll_events.events  = POLLIN | POLLERR;           // 수신된 자료가 있는지, 에러가 있는지
47 poll_events.revents = 0;
48
49
50 // 자료 송수신
51
52 while ( 1)
53 {
54     poll_state = poll(                               // poll()을 호출하여 event 발생 여부 확인
55                     (struct pollfd*)&poll_events, // event 등록 변수
56                     1, // 체크할 pollfd 개수
57                     1000 // time out 시간
58                     );
59
60     if ( 0 < poll_state)                               // 발생한 event 가 있음
61     {
62         if ( poll_events.revents & POLLIN)             // event 가 자료 수신?
63         {
64             cnt = read( fd, buf, 1024);
65             write( fd, buf, cnt);
66             printf( "data received - %d %s\n", cnt, buf);
67         }
68         if ( poll_events.revents & POLLERR)           // event 가 에러?
69         {
70             printf( "통신 라인에 에러가 발생, 프로그램 종료");
71             break;
72         }
73     }
74 }
75 close( fd);
76 return 0;
77 }

```

- 컴파일한 후 실행하면 아래와 같이 나온다 .

```
SSH root@192.168.0.67 (7/3/18 10:17 AM) ✕  
data received - 14 C  
^C  
root@am57xx-evm:~/khj# g++ -o serial serial.cpp $(pkg-config opencv --libs)  
  
data received - 14 ABC  
data received - 14  
data received - 14  
data received - 14 C  
data received - 14 ABC  
data received - 14  
data received - 14  
data received - 14 C
```