



E D D I  
Electronic Design  
Development Institute

---

# 에디로봇아카데미

## 임베디드 마스터 Lv2 과정

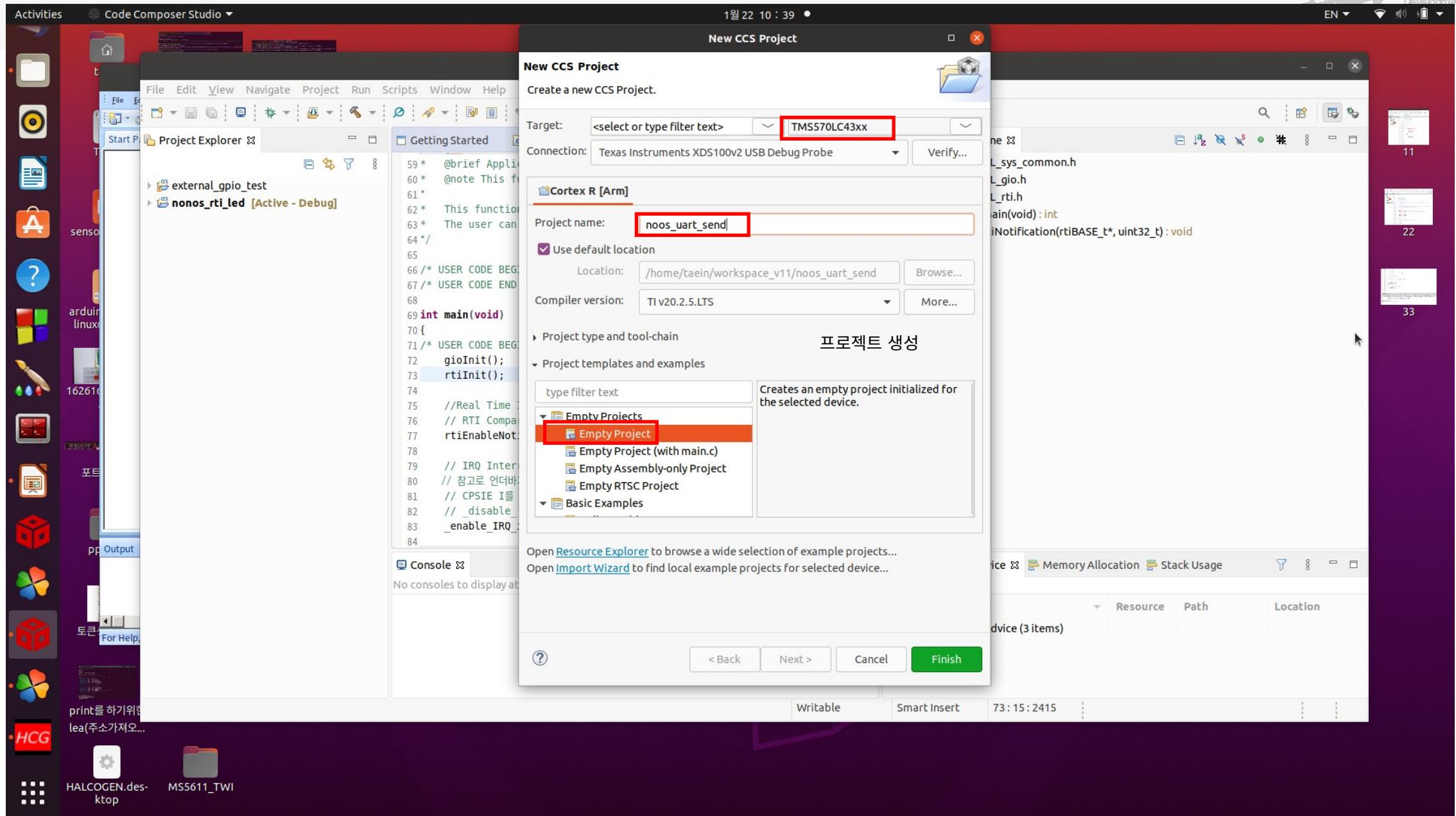
### [TMS 570\_UART(SCI)]

제 1기

2022. 02. 18

박태인

# UART(SCI) – send



# UART(SCI) - send



1월 22 10 : 39 •

workspace\_v11 - nonos\_rti\_led/source/HL\_sys\_main.c - Code Composer Studio

Properties for noos\_uart\_send

Resource

Path: /noos\_uart\_send

Type: Project

Location: /home/taein/workspace\_v11/noos\_uart\_send

Last modified: 2022년 1월 22일 오전 10:39:05

Text file encoding

Inherited from container (UTF-8)

Other: UTF-8

Store the encoding of derived resources separately

New text file line delimiter

Inherited from container (Unix)

Other: Unix

Restore Defaults Apply

Cancel Apply and Close

생성된 프로젝트의  
옵션에서 Location  
복사

Console CDT Build Console [noos\_uart\_send]

Problems Advice Memory Allocation Stack Usage

2 items

Description Resource Path Location

Optimization Advice (2 items)

Activities Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer

- external\_gpio\_test
- nonos\_rti\_led
- noos\_uart\_send [Active - Debug]

Code Editor (HL\_sys\_main.c)

int

71 /\*

72 {

73     BASE\_t\*, uint32\_t) : void

74 }

75

76

77

78

79

80

81

82

83

84

Output

Port

PP

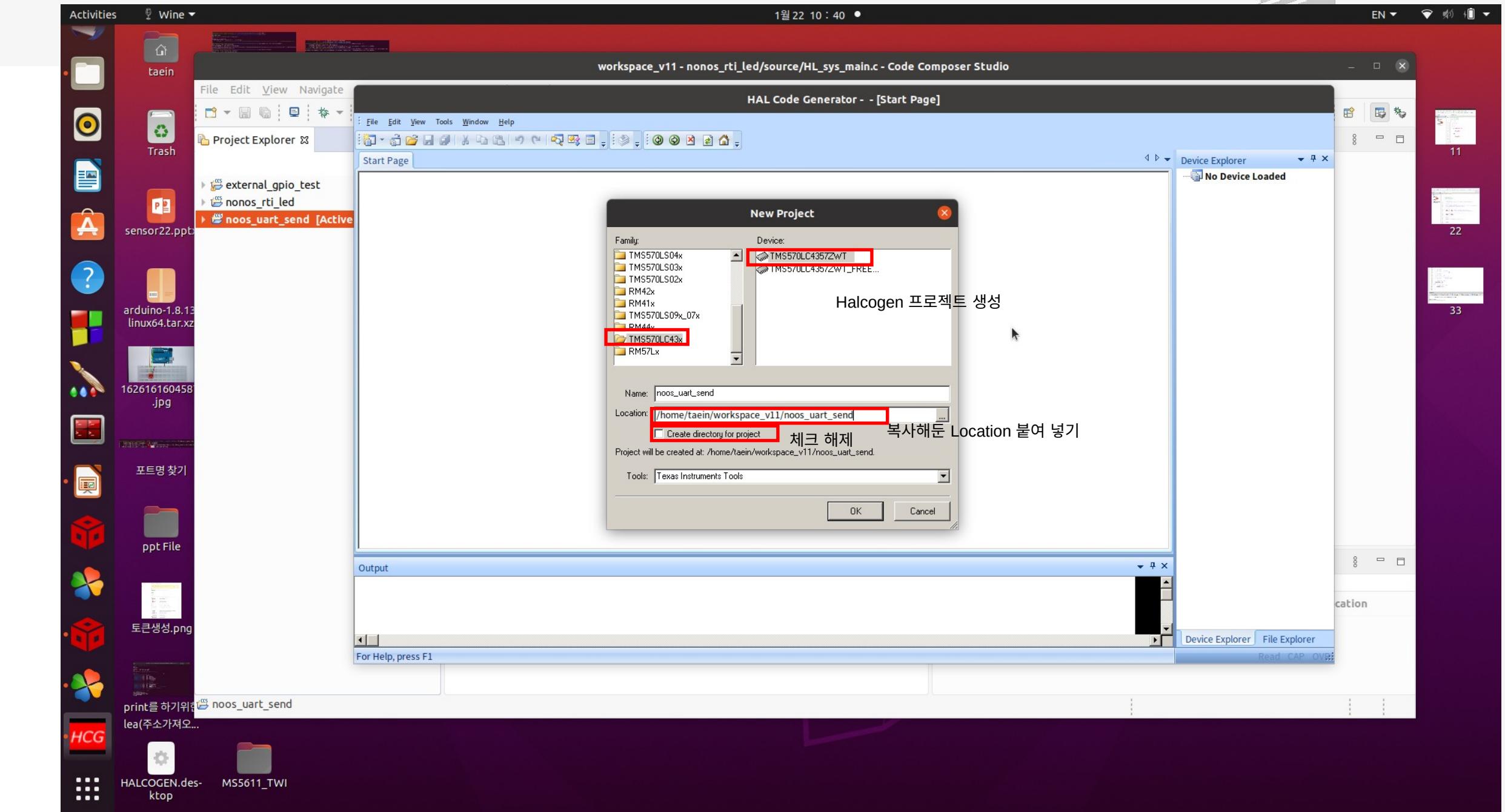
Token

Help

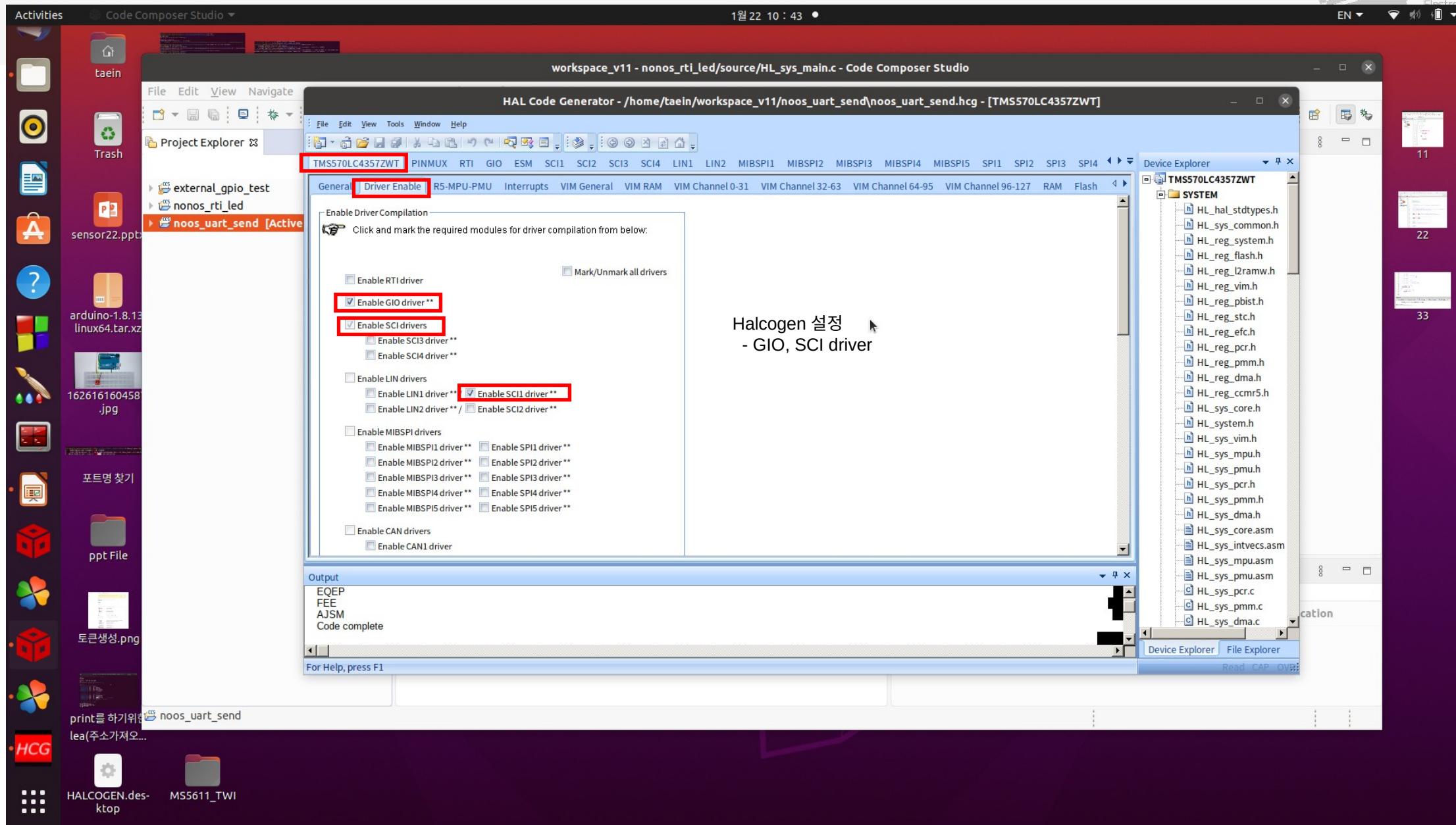
HALCOGEN.desktop MS5611\_TWI

print를 하기위해 noos\_uart\_send  
lea(주소가져오...)

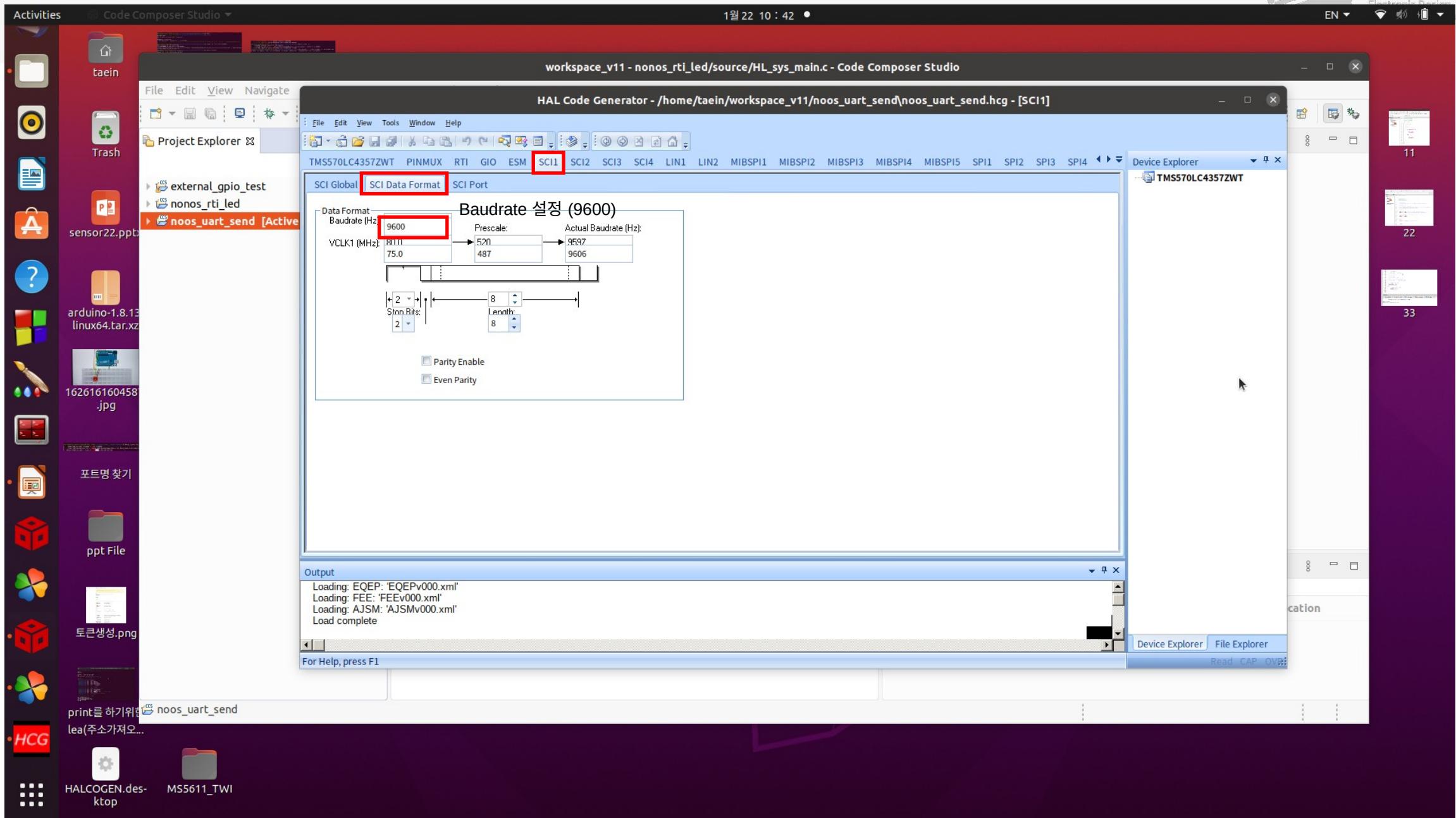
# UART(SCI) - send



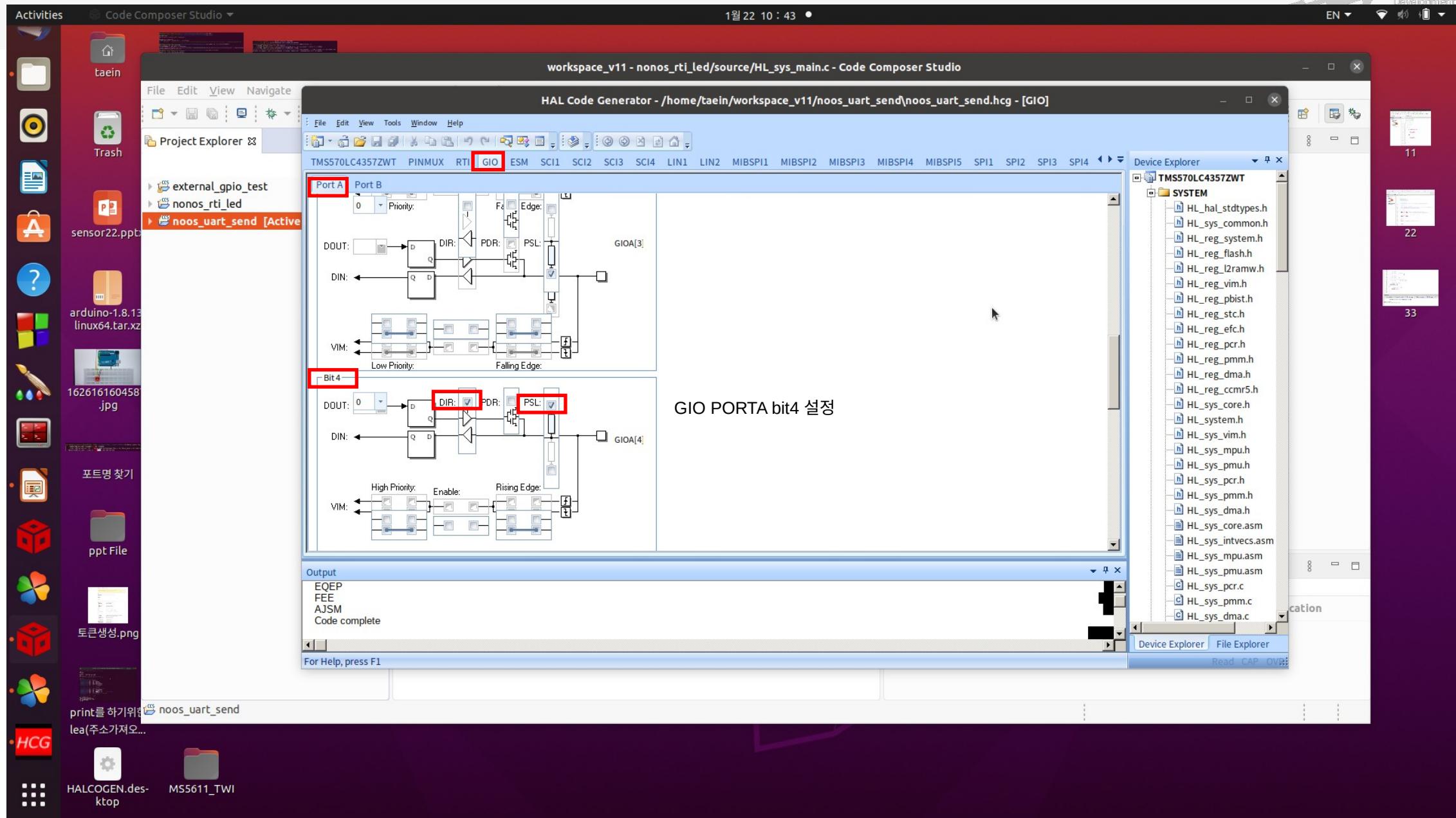
# UART(SCI) - send



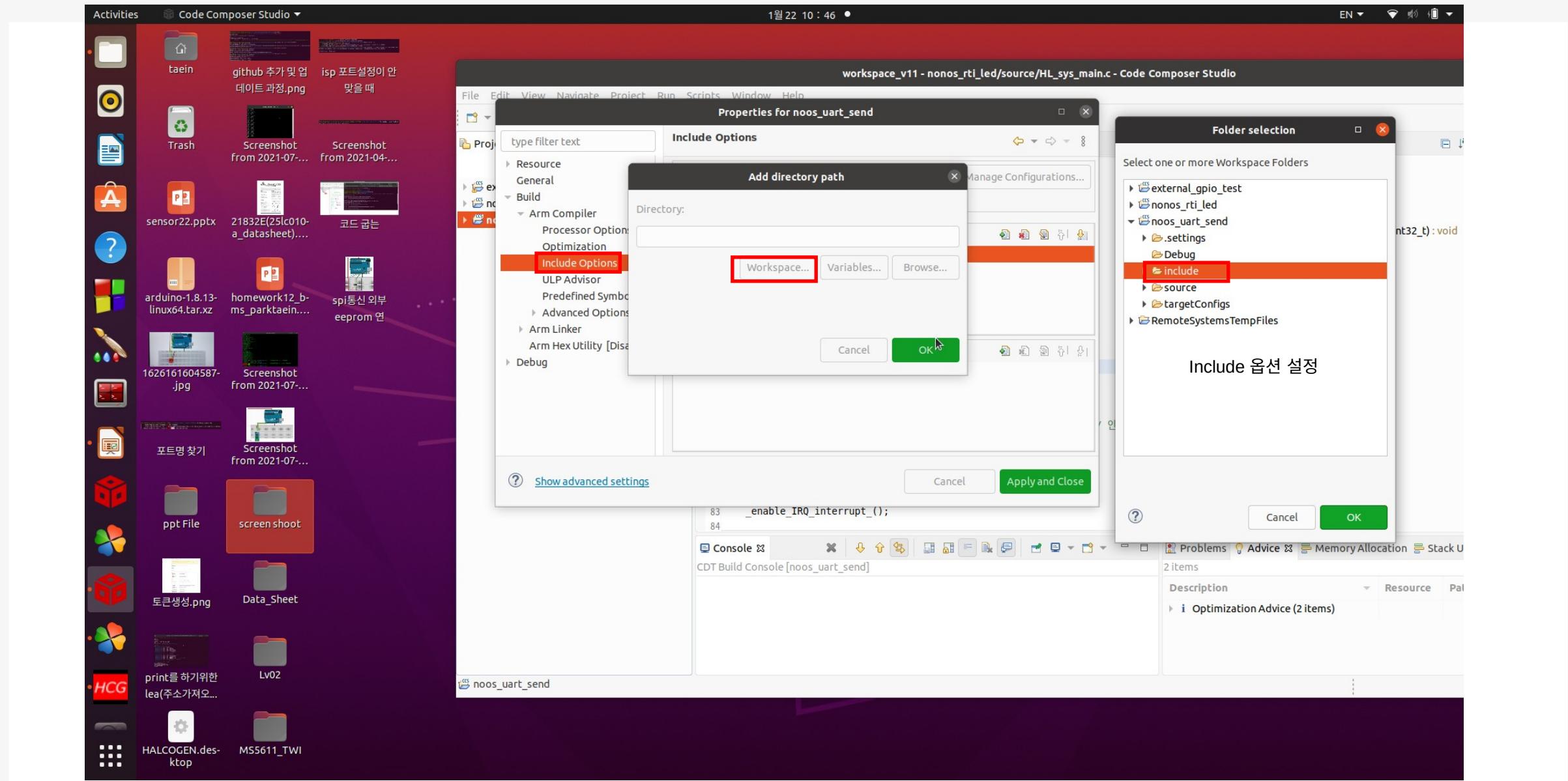
# UART(SCI) - send



# UART(SCI) - send



# UART(SCI) - send



# UART(SCI) - send

Code 분석



workspace\_v11 - noos\_uart\_send/source/HL\_sys\_main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer HL\_sys\_main.c HL\_sys\_main.c HL\_sys\_main.c HL\_sci.c HL\_reg\_sci.h 17

Getting Started HL\_sys\_main.c HL\_sys\_main.c HL\_sci.c HL\_reg\_sci.h 17

42 /\*  
43  
44  
45 /\* USER CODE BEGIN (0) \*/  
46 /\* USER CODE END \*/  
47  
48 /\* Include Files \*/  
49  
50 #include "HL\_sys\_common.h"  
51 /\* USER CODE BEGIN (1) \*/  
52 #include "HL\_system.h"  
53 #include "HL\_sci.h"  
54 #include "HL\_gio.h"  
55 /\* USER CODE END \*/  
56  
57 /\* @fn void main(void)  
58 \* @brief Application main function  
59 \* @note This function is empty by default.  
60 \*  
61 \* This function is called after startup.  
62 \* The user can use this function to implement the application.  
63 \*/  
64  
65  
66 /\* USER CODE BEGIN (2) \*/  
67 #define TSIZE1 6  
68 #define TSIZE2 7  
69 #define TSIZE3 9  
70  
71 uint8 TEXT1[TSIZE1] = { 'E', 'D', 'D', 'I', ' ', ' '};  
72 uint8 TEXT2[TSIZE2] = { 'R', '0', 'B', '0', 'T', ' ', ' '};  
73 uint8 TEXT3[TSIZE3] = { 'A', 'C', 'A', 'D', 'E', 'M', 'Y', '\n', '\r'};

Header 선언 (sci, gio 등)

Outline HL\_sys\_common.h HL\_system.h HL\_sci.h HL\_gio.h # TSIZE1 # TSIZE2 # TSIZE3 TEXT1 : uint8[] TEXT2 : uint8[] TEXT3 : uint8[] sci\_display\_text(sciBASE\_t\*, uint8\*, uint32) : void wait(uint32) : void # UART main(void) : int sci\_display\_text(sciBASE\_t\*, uint8\*, uint32) : void wait(uint32) : void

Console No consoles to display at this time.

Problems Advice Memory Allocation Stack Usage 3 items Description Resource Path Location Optimization Advice (3 items)

Writable Smart Insert 73:1:2524

# UART(SCI) - send

workspace\_v11 - noos\_uart\_send/source/HL\_sys\_main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer HL\_gio.h HL\_gio.c HL\_sys\_dma.c HL\_sys\_main.c HL\_sys\_main.c 15

Getting Started

noos\_uart\_send [Active - Debug]

- Binaries
- Includes
- Debug
- include
- source
  - HL\_epc.c
  - HL\_errata\_SSWF021\_45.c
  - HL\_errata.c
  - HL\_esm.c
  - HL\_gio.c
  - HL\_nmpu.c
  - HL\_notification.c
  - HL\_pimux.c
  - HL\_sci.c
  - HL\_sys\_core.asm
  - HL\_sys\_dma.c
  - HL\_sys\_intvecs.asm
  - HL\_sys\_link.cmd
  - HL\_sys\_main.c
  - HL\_sys\_mpu.asm
  - HL\_sys\_pcr.c
  - HL\_sys\_phantom.c
  - HL\_sys\_pmm.c
  - HL\_sys\_pmu.asm
  - HL\_sys\_startup.c
  - HL\_sys\_vim.c
  - HL\_system.c
- targetConfigs
  - noos\_uart\_send.dil
  - noos\_uart\_send.hcg

Outline

- HL\_sys\_common.h
- HL\_system.h
- HL\_sci.h
- HL\_gio.h
- TSIZE1
- TSIZE2
- TSIZE3
- TEXT1: uint8[]
- TEXT2: uint8[]
- TEXT3: uint8[]
- sci\_display\_text(sciBASE\_t\*, uint8\*, uint32) : void
- wait(uint32) : void

# UART

- main(void) : int
- sci\_display\_text(sciBASE\_t\*, uint8\*, uint32) : void
- wait(uint32) : void

HL\_sys\_main.c

```
45 /* USER CODE BEGIN (0) */  
46 /* USER CODE END */  
47  
48 /* Include Files */  
49  
50 #include "HL_sys_common.h"  
51  
52 /* USER CODE BEGIN (1) */  
53 #include "HL_system.h"  
54 #include "HL_sci.h"  
55 #include "HL_gio.h"  
56 /* USER CODE END */  
57  
58 /** @fn void main(void)  
59 * @brief Application main function  
60 * @note This function is empty by default.  
61 *  
62 * This function is called after startup.  
63 * The user can use this function to implement the application.  
64 */  
65  
66 /* USER CODE BEGIN (2) */  
67 #define TSIZE1 6  
68 #define TSIZE2 7  
69 #define TSIZE3 9  
70  
71 uint8 TEXT1[TSIZE1] = { 'E', 'D', 'D', 'I', ' ', ' '};  
72 uint8 TEXT2[TSIZE2] = { 'R', 'O', 'B', 'O', 'T', ' ', ' '};  
73 uint8 TEXT3[TSIZE3] = { 'A', 'C', 'A', 'D', 'E', 'M', 'Y', '\n', '\r'};  
74  
75 void sci_display_text (sciBASE_t *sci, uint8 *text, uint32 length);  
76 void wait (uint32 time);
```

Console

No consoles to display at this time.

Problems Advice Memory Allocation Stack Usage

3 items

Description	Resource	Path	Location
Optimization Advice (3 items)			

Writable Smart Insert 74:1:2596

# UART(SCI) - send



**EDDI**  
Electronic Design  
Development Institute

workspace\_v11 - noos\_uart\_send/source/HL\_sys\_main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer

- eCAP (in RC\_controller\_ecap) [EmbeddedMaster]
- ecap\_rf\_control
- external\_gpio\_test
- nonos\_rti\_led
- noos\_etpwm\_control\_with\_uart
- noos\_uart\_receive
- noos\_uart\_send [Active - Debug]
  - Binaries
  - Includes
  - Debug
  - include
  - source
    - HL\_epc.c
    - HL\_errata\_SSWF021\_45.c
    - HL\_errata.c
    - HL\_esm.c
    - HL\_gio.c
    - HL\_nmpu.c
    - HL\_notification.c
    - HL\_pinmux.c
    - HL\_sci.c
    - HL\_sys\_core.asm
    - HL\_sys\_dma.c
    - HL\_sys\_intvecs.asm
    - HL\_sys\_link.cmd
    - HL\_sys\_main.c
    - HL\_sys\_mpu.asm
    - HL\_sys\_pcr.c
    - HL\_sys\_phantom.c
    - HL\_sys\_pmm.c
    - HL\_sys\_pmu.asm
    - HL\_sys\_startup.c
    - HL\_sys\_vim.c
    - HL\_system.c

Getting Started HL\_gio.h HL\_gio.c HL\_sys\_dma.c HL\_sys\_main.c HL\_sys\_main.c 15

75 void sci\_display\_text (sciBASE\_t \*sci, uint8 \*text, uint32 length);  
76 void wait (uint32 time);  
77  
78 #define UART sciREG1  
79 /\* USER CODE END \*/ SciREG1은 UART로 define

사용하게 될 함수 원형 선언  
(send하게 되는 것이고)

81 int main(void)  
82 {  
83 /\* USER CODE BEGIN (3) \*/  
84 gioInit();  
85 sciInit();  
86  
87 //gioSetDirection(gioPORTA, 0xffffffff);  
88 for(;)  
89 {  
90 sci\_display\_text(UART, &TEXT1[0], TSIZE1);  
91 sci\_display\_text(UART, &TEXT2[0], TSIZE2);  
92 sci\_display\_text(UART, &TEXT3[0], TSIZE3);  
93  
94 gioToggleBit(gioPORTA, 4);  
95  
96 wait(50000000);  
97 }  
98 /\* USER CODE END \*/  
99  
100 return 0;  
101}  
102  
103  
104 /\* USER CODE BEGIN (4) \*/  
105 void sci\_display\_text(sciBASE\_t \*sci, uint8 \*text, uint32 length)

Sci\_display\_text 함수를 활용하는 모습  
(sciREG1 선택,  
TEXT1~3중에 처음 위치 [0]을 가르키는 포인터  
TEXT의 사이즈 )

함수 원형

Outline

- HL\_sys\_common.h
- HL\_system.h
- HL\_sci.h
- HL\_gio.h
- # TSIZE1
- # TSIZE2
- # TSIZE3
- TEXT1 : uint8[]
- TEXT2 : uint8[]
- TEXT3 : uint8[]
- + sci\_display\_text(sciBASE\_t\*, uint8\*, uint32) : void
- + wait(uint32) : void
- # UART
- main(void) : int
- sci\_display\_text(sciBASE\_t\*, uint8\*, uint32) : void
- wait(uint32) : void

/\* USER CODE BEGIN (4) \*/  
void sci\_display\_text(sciBASE\_t \*sci, uint8 \*text, uint32 length)  
{  
while(length--)  
{  
while ((UART->FLR & 0x4) == 4) // 조건문에 해당되면 busy 하다는 얘기  
;  
sciSendByte(UART, \*text++); // 1 바이트씩 움직일 거니까 포인터로

uint32 FLR; /\*< 0x001C Interrupt Flag Register \*/

질문?? 여기서 4 도 0x4의 의미로 보면 되는 건가요?

29.7.8 SCI Flags Register (SCIFLR)

Figure 29-35 and Table 29-20 illustrate this register.

Figure 29-35. SCI Flags Register (SCIFLR) (offset = 1Ch)

31	30	29	28	27	26	25	24
BE	PBE	CE	ISFE	NRE	FE	OE	PE
R/WL-0	R/WL-0	R/WL-0	R/WL-0	R/WL-0	R/W-0	R/W-0	R/W-0
23							16
Reserved							
15	14	13	12	11	10	9	8
Reserved	ID RX	ID TX	RX WAKE	R/WC-0	R/W-1	R/WC-0	R/W-0
R-0	R/WL-0	R/WL-0	R/WC-0	R/W-1	R/WC-0	R/W-0	R/W-1
7	6	5	4	3	2	1	0
TOA3WUS	TOAWUS	Reserved	TIMEOUT	BUSY	idle	WAKE UP	BRKDT
R/WL-0	R/WL-0	R-0	R/WL-0	R/W-0	R-0	R/WL-0	R/WC-0

LEGEND: R/W = Read/Write; R = Read only; WC = Write in SCI-compatible mode only; WL = Write in LIN mode only. -n = value after reset

Problems Advice Memory Allocation Stack Usage

ems

i Opti 3 BUSY

Bus busy flag. This bit is effective in LIN mode and SCI-compatible mode. This bit indicates whether the receiver is in the process of receiving a frame. As soon as the receiver detects the beginning of a start bit, the BUSY bit is set to 1. When the reception of a frame is complete, the SCI/LIN clears the BUSY bit. If SET WAKEUP INT is set and power down is requested while this bit is set, the SCI/LIN automatically prevents low-power mode from being entered and generates wake-up interrupt. The BUSY bit is controlled directly by the SCI/LIN receiver, but this bit can also be cleared by the following:

- Setting the SWnRST bit
- Setting of the RESET bit
- A system reset occurring

0 The receiver is not currently receiving a frame.  
1 The receiver is currently receiving a frame.

오늘 1이면 무어가 스시 즈이다 라고 한다

# UART(SCI) - send

workspace\_v11 - noos\_uart\_send/source/HL\_sys\_main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer HL\_gio.h HL\_gio.c HL\_sys\_main.c HL\_sys\_main.c HL\_sci.c 16

Getting Started eCAP (in RC\_controller\_ecap) [EmbeddedMaster] ecap\_rf\_control external\_gpio\_test nonos\_rti\_led noos\_etpwm\_control\_with\_uart noos\_uart\_recieve noos\_uart\_send [Active - Debug] Binaries Includes Debug include source HL\_epc.c HL\_errata\_SSWF021\_45.c HL\_errata.c HL\_esm.c HL\_gio.c HL\_nmpu.c HL\_notification.c HL\_pinmux.c HL\_sci.c HL\_sys\_core.asm HL\_sys\_dma.c HL\_sys\_intvecs.asm HL\_sys\_link.cmd HL\_sys\_main.c HL\_sys\_mpu.asm HL\_sys\_pcr.c HL\_sys\_phantom.c HL\_sys\_pmm.c HL\_sys\_pmu.asm HL\_sys\_startup.c HL\_sys\_vim.c HL\_system.c targetConfigs noos\_uart\_send.dil noos\_uart\_send.hcg

sci\_display\_text(sciBASE\_t \*sci, uint8 \*text, uint32 length)

105 void sci\_display\_text(sciBASE\_t \*sci, uint8 \*text, uint32 length)

106 {

107 while(length--)

108 {

109 while ((UART->FLR & 0x4) == 4) // 조건문에 해당되면 busy 하다는 얘기

110 ;

111 sciSendByte(UART, \*text++); // 1 바이트씩 움직일 거니까 포인터로

112 }

113 }

114 }

115

116 void wait(uint32 time)

117 {

118 int i;

119 for(i=0; i<time; i++)

120 ;

121 }

122 }

123 /\* USER CODE END \*/

124

sciSendByte(sciBASE\_t \*sci, uint8 byte)

251 void sciSendByte(sciBASE\_t \*sci, uint8 byte)

252 {

253 /\* USER CODE BEGIN (9) \*/

254 /\* USER CODE END \*/

255 /\*SAFETYMCUSW 28 D MR:NA <APPROVED> "Potentially infinite loop found - Hardware Status check for execution sequence."\*/

256 while ((sci->FLR & (uint32)SCI\_TX\_INT) == 0U)

257 {

258 /\* Wait \*/

259 sci->TD = byte;

260 }

261 /\* USER CODE BEGIN (10) \*/

262 /\* USER CODE END \*/

263

87 uint32 TD; /\*\*< 0x0038 Transmit Data Buffer \*/

Console No consoles to display at this time.

Problems Advice Memory Allocation Stack Usage

3 items

Description Resource Path Location

i Optimization Advice (3 items)

Writable Smart Insert 98:20:3136

잠재적인 무한 루프 발견??  
OU라는건 0000을 의미?  
Busy 하지 않은 때, Data 전송

# UART(SCI) - send

ScInit 분석



```
void sciInit(void)
{
/* USER CODE BEGIN (2) */
/* USER CODE END */

/** @b initialize @b SCI1 */

/** - bring SCI1 out of reset */
sciREG1->GCR0 = 0U;
sciREG1->GCR0 = 1U;
```

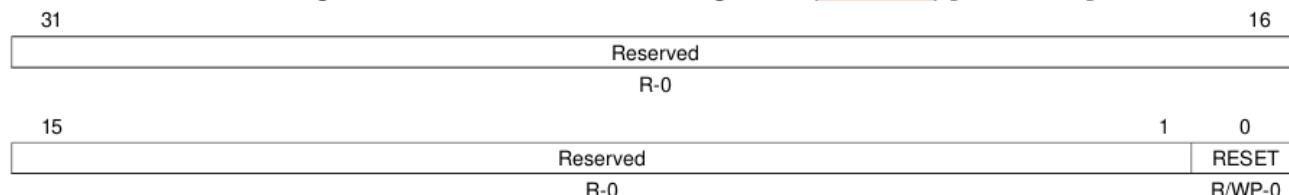
Table 29-10. SCI/LIN Control Registers

Offset	Acronym	Register Description	Section
00h	SCIGCR0	SCI Global Control Register 0	Section 29.7.1

## 30.7.1 SCI Global Control Register 0 (SCIGCR0)

The SCIGCR0 register defines the module reset. [Figure 30-8](#) and [Table 30-4](#) illustrate this register.

Figure 30-8. SCI Global Control Register 0 (SCIGCR0) [offset = 00]



LEGEND: R/W = Read/Write; R = Read only; R/WP = Read/Write in privileged mode only; -n = value after reset

Table 30-4. SCI Global Control Register 0 (SCIGCR0) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reads return 0. Writes have no effect.
0	RESET	0	This bit resets the SCI module.
		1	SCI module is in reset.
			SCI module is out of reset.
			<b>Note: Read/Write in privileged mode only.</b>

# UART(SCI) - send

ScInit 분석



```
/** - Disable all interrupts */
sciREG1->CLEARINT = 0xFFFFFFFFU;           uint32 CLEARINT;      /**< 0x0010 Clear Interrupt Enable Register */
sciREG1->CLEARINTLVL = 0xFFFFFFFFU;          uint32 CLEARINTLVL;   /**< 0x0018 Set Interrupt Level Register */
```

**Table 29-17. SCI Clear Interrupt Register (SCICLEARINT) Field Descriptions (continued)**

4	CLR TIMEOUT INT	0	Clear timeout interrupt. This bit is effective in LIN mode only. This bit disables the timeout (LIN bus idle) interrupt when set. <i>Read:</i> The interrupt is disabled. <i>Write:</i> No effect.
		1	<i>Read:</i> The interrupt is enabled. <i>Write:</i> The interrupt is disabled.

## 29.7.7 SCI Clear Interrupt Level Register (SCICLEARINTLVL)

Figure 29-34 and Table 29-19 illustrate this register.

**Figure 29-34. SCI Clear Interrupt Level Register (SCICLEARINTLVL) (offset = 18h)**

31	30	29	28	27	26	25	24
CLR BE INT LVL	CLR PBE INT LVL	CLR CE INT LVL	CLR ISFE INT LVL	CLR NRE INT LVL	CLR FE INT LVL	CLR OE INT LVL	CLR PE INT LVL
R/WL-0	R/WL-0	R/WL-0	R/WL-0	R/WL-0	R/W-0	R/W-0	R/W-0
23			19	18	17	16	
Reserved				CLR RX DMA ALL INT LVL	Reserved		
			R-0	R/WC-0		R-0	
15	14	13	12	10	9	8	
Reserved		CLR ID INT LVL	Reserved		CLR RX INT LVL	CLR TX INT LVL	
R-0		R/WL-0	R-0		R/W-0	R/W-0	
7	6	5	4	3	2	1	0
CLR TOA3WUS INT LVL	CLR TOAWUS INT LVL	Reserved	CLR TIMEOUT INT LVL	Reserved	CLR WAKEUP INT LVL	CLR BRKDT INT LVL	
R/WL-0	R/WL-0	R-0	R/WL-0	R-0	R/W-0	R/WC-0	

LEGEND: R/W = Read/Write; R = Read only; WL = Write in LIN mode only; WC = Write in SCI-compatible mode only; -n = value after reset

4	CLR TIMEOUT INT LVL	0	Clear timeout interrupt. This bit is effective in LIN mode only. <i>Read:</i> The interrupt level is mapped to the INT0 line. <i>Write:</i> No effect.
		1	<i>Read:</i> The interrupt level is mapped to the INT1 line. <i>Write:</i> The interrupt level is mapped to the INT0 line.
3-2	Reserved	0	Reads return 0. Writes have no effect.

1이면 어떤 효과??

# UART(SCI) - send

Scilnit 분석



```
/** - global control 1 */
sciREG1->GCR1 = (uint32)((uint32)1U << 25U) /* enable transmit */
| (uint32)((uint32)1U << 24U) /* enable receive */
| (uint32)((uint32)1U << 5U) /* internal clock (device has no clock pin) */
| (uint32)((uint32)(2U-1U) << 4U) /* number of stop bits */
| (uint32)((uint32)0U << 3U) /* even parity, otherwise odd */
| (uint32)((uint32)0U << 2U) /* enable parity */
| (uint32)((uint32)1U << 1U); /* asynchronous timing mode */

uint32 GCR1;          /**< 0x0004 Global Control Register 1 */
```

Table 30-5. SCI Global Control Register 1 (SCIGCR1) Field Descriptions

2	PARITY ENA	0	Parity enable. This bit enables or disables the parity function.
		1	Parity is disabled; no parity bit is generated during transmission or is expected during reception.

# UART(SCI) - send

## ScInit 분석

```
/** - set baudrate */
sciREG1->BRS = 487U; /* baudrate */

/** - transmission length */
sciREG1->FORMAT = 8U - 1U; /* length */ 8U – 1U 란??
```

Table 29-24. Baud Rate Selection Register (BRS) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reads return 0. Writes have no effect.
30-28	U	0-2h	SCI/LIN super fractional divider selection. These bits are effective in LIN or SCI asynchronous mode. These bits are an additional fractional part for the baud rate specification. These bits allow a super-fine tuning of the fractional baud rate with seven more intermediate values for each of the M fractional divider values. See <a href="#">Section 29.3.1.4.1</a> for more details.
27-24	M	0-3h	SCI/LIN 4-bit fractional divider selection. These bits are effective in LIN or SCI asynchronous mode. These bits are used to select a baud rate for the SCI/LIN module, and they are a fractional part for the baud rate specification. The M divider allows fine-tuning of the baud rate over the P prescaler with 15 additional intermediate values for each of the P integer values. See <a href="#">Section 29.3.1.4.1</a> for more details.
23-0	PRESCALER P	0-FF FFFFh	<p>These bits are used to select a baud rate for the SCI/LIN module. These bits are effective in LIN mode and SCI compatibility.</p> <p>The SCI/LIN has an internally generated serial clock determined by the VCLK and the prescalers P and M in this register. The LIN uses the 24-bit integer prescaler P value of this register to select one of over 16,700,000. The additional 4-bit fractional divider M refines the baud rate selection PRESCALER[27:24].</p> <p><b>NOTE: In LIN mode, ONLY the asynchronous mode and baud rate values are used.</b></p> <p>The baud rate can be calculated using the following formulas:</p> $\text{Asynchronous baud value} = \left( \frac{\text{VCLK Frequency}}{16(P + 1 + \frac{M}{16})} \right) \quad (52)$ $\text{Isynchronous baud value} = \left( \frac{\text{VCLK Frequency}}{P + 1} \right) \quad (53)$ <p>For P = 0,</p> $\text{Asynchronous baud value} = \left( \frac{\text{VCLK Frequency}}{32} \right) \quad (54)$ $\text{Isynchronous baud value} = \left( \frac{\text{VCLK Frequency}}{2} \right) \quad (55)$ <p>Table 29-25 contains comparative baud values for different P values, with VCLK = 50 MHz, for asynchronous mode.</p>

```
uint32 BRS;           /**< 0x002C Baud Rate Selection Register */
uint32 FORMAT;        /**< 0x0028 Format Control Register */
```



Table 29-25. Comparative Baud Values for Different P Values, Asynchronous Mode<sup>(1)(2)</sup>

24-Bit Register Value		Baud Selected		Percent Error
Decimal	Hex	Ideal	Actual	
26	00001A	115200	115740	0.47
53	000035	57600	57870	0.47
80	000050	38400	38580	0.47
162	0000A2	19200	19172	-0.15
299	00012B	10400	10417	0.16
325	000145	9600	9586	-0.15
399	00018F	7812.5	7812.5	0.00
650	00028A	4800	4800	0.00
15624	003BA0	200	200	0.00
624999	098967	5	5	0.00

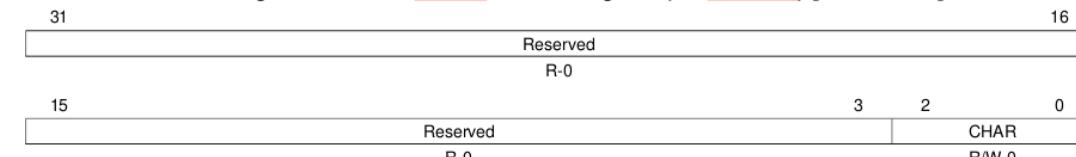
<sup>(1)</sup> VCLK = 50 MHz

<sup>(2)</sup> Values are in decimal except for column 2.

## 30.7.10 SCI Format Control Register (SCI FORMAT)

Figure 30-17 and [Table 30-15](#) illustrate this register.

Figure 30-17. SCI Format Control Register (SCI FORMAT) [offset = 28h]



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 30-15. SCI Format Control Register (SCI FORMAT) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reads return 0. Writes have no effect.
2-0	CHAR		<p>Character length control bits. These bits set the SCI character length from 1 to 8 bits. When data of fewer than eight bits in length is received, it is left-justified in SCIRD and padded with trailing zeros. Data read from the SCIRD should be shifted by software to make the received data right-justified. Data written to the SCITD should be right-justified but does not need to be padded with leading zeros.</p> <p>0 The character is 1 bit long. 1h The character is 2 bits long. 2h The character is 3 bits long. 3h The character is 4 bits long. 4h The character is 5 bits long. 5h The character is 6 bits long. 6h The character is 7 bits long. 7h The character is 8 bits long.</p>

# UART(SCI) - send

Activities Terminator 1월 22 11 : 04 workspace\_v11 - noos\_uart\_send/source/HL\_sys\_main.c - Code Composer Studio

taein@taein-Lenovo-ideapad-700-15ISK: ~ taein@taein-Lenovo-ideapad-700-15ISK: ~ 80x24 taein@taein-Lenovo-ideapad-700-15ISK:~\$ putty & [red box]

HL\_sys\_main.c HL\_sys\_main.c 88

```
    , ' ', ' ');
    , ' ', ' ');
    E, 'M', 'Y', '\n', '\r');

    *text, unit32 length);

SCI 통신 확인을 위한
Putty 실행.
```

HL\_sys\_dma.c HL\_sys\_intvecs.asm HL\_sys\_link.cmd HL\_sys\_main.c HL\_sys\_mpu.asm HL\_sys\_pcr.c HL\_sys\_phantom.c HL\_sys\_pmm.c HL\_sys\_pmum.asm HL\_sys\_startup.c HL\_sys\_vim.c HL\_system.c targetConfigs noos\_uart\_send.dil noos\_uart\_send.hcg

86 //gioSetDirection(gioPORTA, 0xffffffff);
87 for(;;)
88 {
89 sci\_display\_text(UART, &TEXT1[0], TSIZE1);
90 sci\_display\_text(UART, &TEXT2[0], TSIZE2);
91 sci\_display\_text(UART, &TEXT3[0], TSIZE3);
92
93 gioToggleBit(gioPORTA, 4);
94
95 wait(50000000);
96 }
97 /\* USER\_CODE\_END \*/
98

Console CDT Build Console [noos\_uart\_send]
diag\_warning=225 --diag\_wrap=off --display\_error\_number --enum\_type=packed --abi=eabi --
preproc\_with\_compile --preproc\_dependency="source/HL\_sys\_vim.d\_raw" --
obj\_directory="source" "../source/HL\_sys\_vim.c"
Finished building: "../source/HL\_sys\_vim.c"

gmake: Target 'all' not remade because of errors.

\*\*\*\* Build Finished \*\*\*\*

Problems Advice Memory Allocation Stack Usage
8 errors, 3 warnings, 0 others

Description	Resource	Path	Location
Errors (8 items)			
#20 identifier "TSIZE2" is undefined	HL_sys_mair	/noos_uart_send	line 72
#20 identifier "TSIZE3" is undefined	HL_sys_mair	/noos_uart_send	line 73
#20 identifier "unit32" is undefined	HL_sys_mair	/noos_uart_send	line 75
#20 identifier "unit32" is undefined	HL_sys_mair	/noos_uart_send	line 105
#20 identifier "unit32" is undefined	HL_sys_mair	/noos_uart_send	line 116

Errors (8 items)

# UART(SCI) - send



1월 22 11:10 ● workspace\_v11 - noos\_uart\_send/source/HL\_sys\_main.c - Code Composer Studio

taein@taein-Lenovo-ideapad-700-15ISK: ~

```
taein@taein-Lenovo-ideapad-700-15ISK: ~ 80x24
[1] 10428
[putty:10428]: Gtk-CRITICAL **: 11:04:05.950: gtk_box_gadget_distribute: assertion 'size >= 0' failed in GtkScrollbar
[putty:10428]: Gtk-CRITICAL **: 11:04:05.954: gtk_box_gadget_distribute: assertion 'size >= 0' failed in GtkScrollbar
[putty:10428]: Gtk-CRITICAL **: 11:04:05.956: gtk_box_gadget_distribute: assertion 'size >= 0' failed in GtkScrollbar
```

File HL\_sys\_main.c HL\_sys\_main.c Outline

PUTTY Configuration

Session

Specify the destination you want to connect to

Serial line: /dev/ttyUSB0

Speed: 9600

Connection type: Serial

Logging Terminal Keyboard Bell Features Window

Load, save or delete a stored session

Saved Sessions

Load Save Delete

Only on clean exit

Open Cancel

taein@taein-Lenovo-ideapad-700-15ISK: ~

```
taein@taein-Lenovo-ideapad-700-15ISK: ~ 80x24
ask="r" denied_mask="r" fsuid=1000 ouid=1000
[ 896.611924] audit: type=1400 audit(1642814189.538:59): apparmor="ALLOWED" operation="file_perm" profile="libreoffice-oopsplash" name="/tmp/OSL_PIPE_1000_SingleOfficeIPC_aee16bc15eb1f93adb6b469f92a4d33" pid=7511 comm="oopsplash" requested_m
ask="r" denied_mask="r" fsuid=1000 ouid=1000
[ 896.611928] audit: type=1400 audit(1642814189.538:60): apparmor="ALLOWED" operation="file_perm" profile="libreoffice-oopsplash" name="/tmp/OSL_PIPE_1000_SingleOfficeIPC_aee16bc15eb1f93adb6b469f92a4d33" pid=7511 comm="oopsplash" requested_m
ask="w" denied_mask="w" fsuid=1000 ouid=1000
[ 896.611931] audit: type=1400 audit(1642814189.538:61): apparmor="ALLOWED" operation="file_perm" profile="libreoffice-oopsplash" name="/tmp/OSL_PIPE_1000_SingleOfficeIPC_aee16bc15eb1f93adb6b469f92a4d33" pid=7511 comm="oopsplash" requested_m
ask="w" denied_mask="w" fsuid=1000 ouid=1000
taein@taein-Lenovo-ideapad-700-15ISK: ~$ ls
Arduino snap 문서
PlayOnLinux's virtual drives sw 바탕화면
Smart-City tt 비디오
arduino-1.8.13 winehq.key 사진
dfu-util-0.10 workspace_v11 음악
dfu-util-0.10.tar.gz 다운로드 템플릿
proj
taein@taein-Lenovo-ideapad-700-15ISK: ~$ ls /dev/ttys
/dev/ttys
```

Problems Advice Memory Allocation Stack Usage

Resource Path Location

Warnings (1 item)

#112-D statement is unreachable

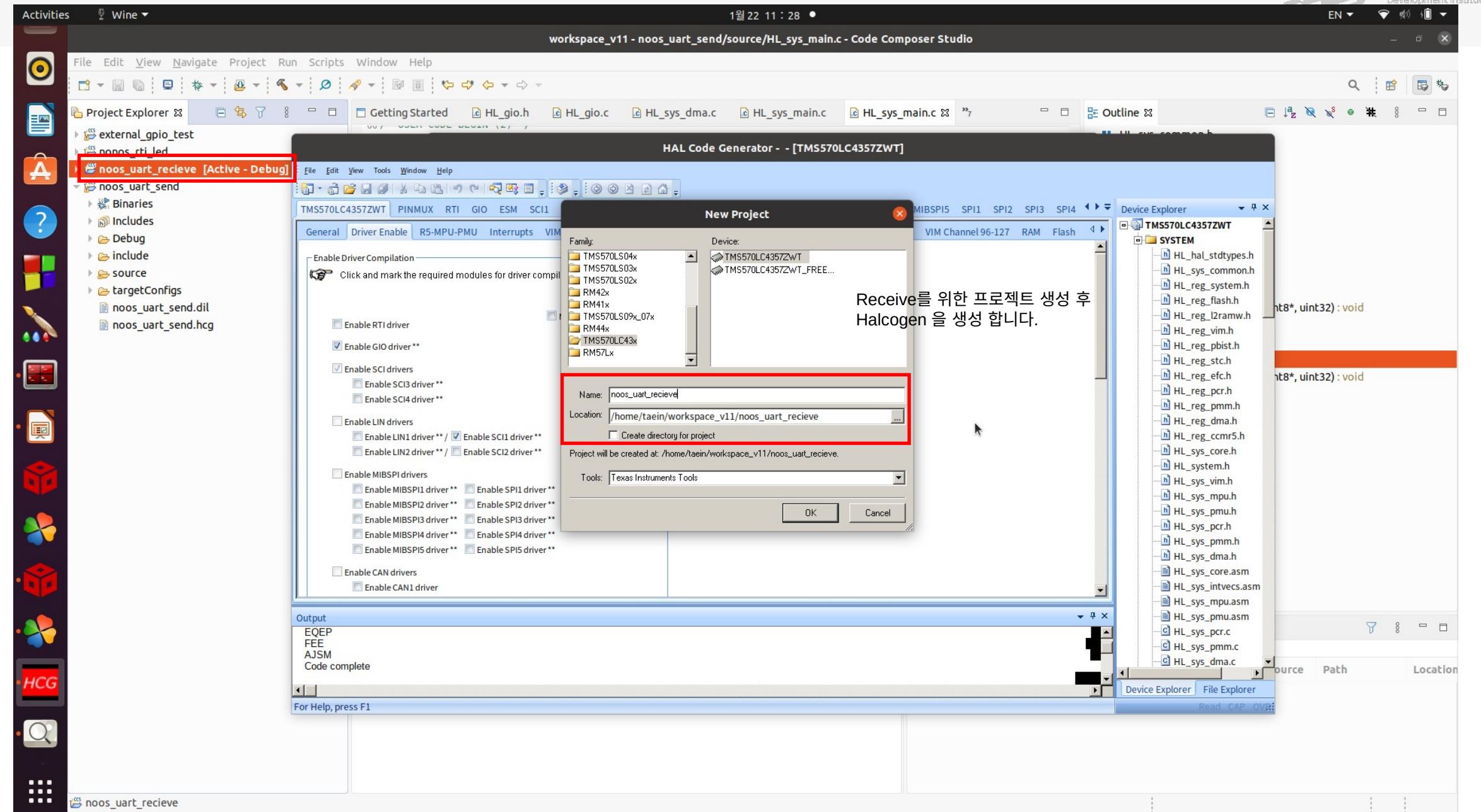
HL\_sys\_main /noos\_uart\_send line 100

Putty 연결 시 포트 확인 방법

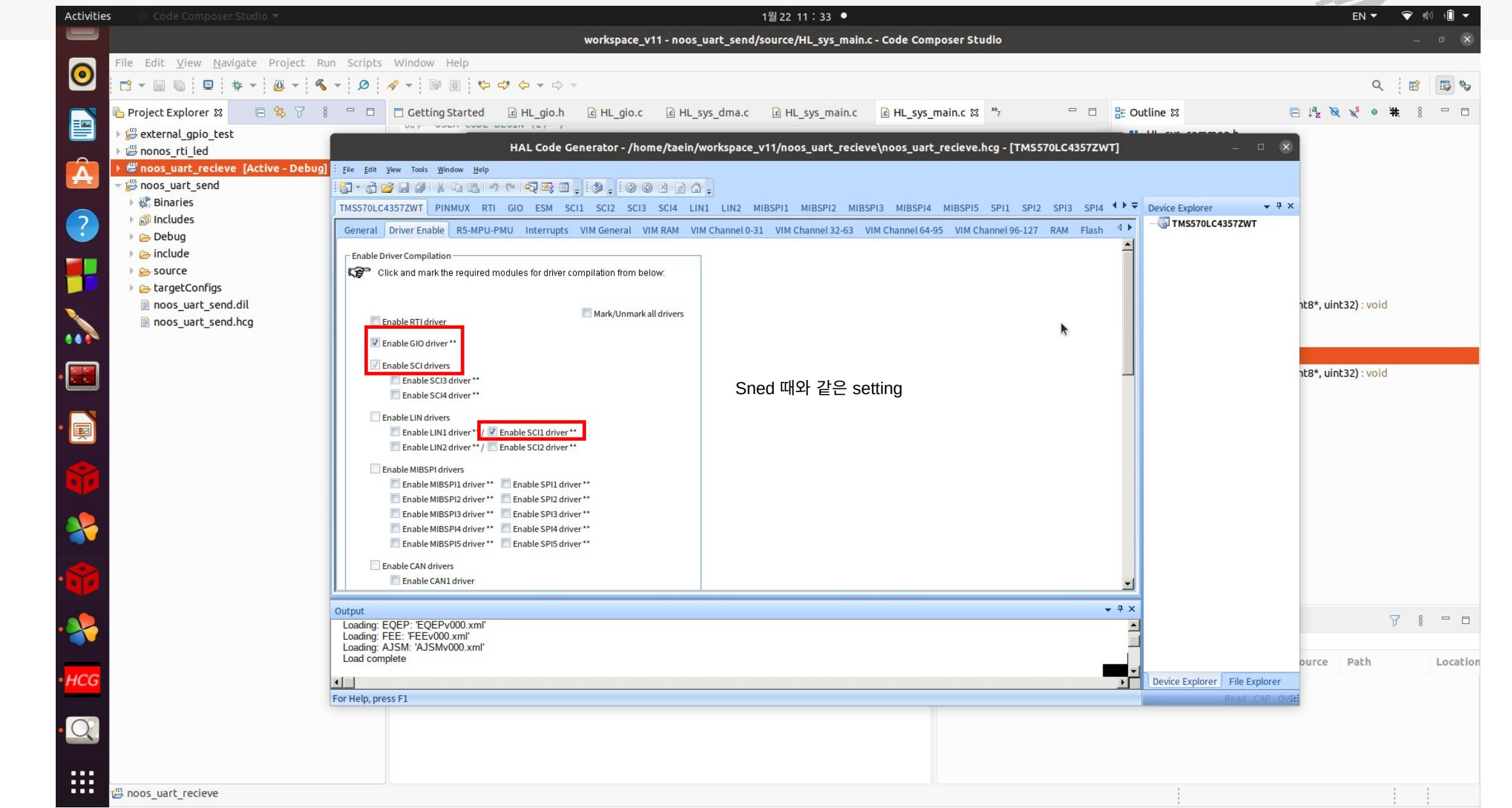
\*\*\*\* Build Finished \*\*\*\*

Writable Smart Insert 109:17:3303

# UART(SCI) – receive



# UART(SCI) – receive



# UART(SCI) – receive



EDDI

Electronic Design  
Development Interface

EN

Activities Code Composer Studio • 1월 22 11 : 33 • workspace\_v11 - noos\_uart\_send/source/HL\_sys\_main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer Getting Started HL\_gio.h HL\_gio.c HL\_sys\_dma.c HL\_sys\_main.c HL\_sys\_main.c Outline

external\_gpio\_test nonos\_rti\_led noos\_uart\_recieve [Active - Debug] noos\_uart\_send Binaries Includes Debug include source targetConfigs noos\_uart\_send.dil noos\_uart\_send.hcg

HAL Code Generator - /home/taein/workspace\_v11/noos\_uart\_recieve/noos\_uart\_recieve.hcg - [GIO]

TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3 MIBSPI4 MIBSPI5 SPI1 SPI2 SPI3 SPI4 Port A Port B

VIM: Low Priority: Falling Edge: Bit 4 DIR: PDR: PSL:

DOUT: 0 D DIN: GIOA[4]

High Priority: Enable: Rising Edge: VIM: Bit 5 DIR: PDR: PSL:

DOUT: 0 D DIN: GIOA[5]

GIO pull-up 설정

Output

Loading: EQEP: 'EQEPv000.xml'  
Loading: FEE: 'FEEv000.xml'  
Loading: AJSM: 'AJSMv000.xml'  
Load complete

Device Explorer File Explorer Read CAP OVER

noos\_uart\_recieve

# UART(SCI) – receive

workspace\_v11 - noos\_uart\_recieve/source/HL\_sys\_main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer HL\_gio.h HL\_sys\_main.c HL\_sys\_main.c HL\_sci.c 16

Getting Started HL\_gio.h HL\_sys\_main.c HL\_sys\_main.c HL\_sci.c 16

noos\_uart\_recieve [Active - Debug]

Binaries Includes Debug include source HL\_epc.c HL\_errata\_SSWF021\_45.c HL\_errata.c HL\_esm.c HL\_gio.c HL\_nmpu.c HL\_notification.c HL\_pinmux.c HL\_sci.c HL\_sys\_core.asm HL\_sys\_dma.c HL\_sys\_intvecs.asm HL\_sys\_link.cmd HL\_sys\_main.c HL\_sys\_mpu.asm HL\_sys\_pcr.c HL\_sys\_phantom.c HL\_sys\_pmm.c HL\_sys\_pmu.asm HL\_sys\_startup.c HL\_sys\_vim.c HL\_system.c targetConfigs noos\_uart\_recieve.dil noos\_uart\_recieve.hcg noos\_uart\_send

51 /\* USER CODE BEGIN (1) \*/  
52 #include "HL\_system.h"  
53 #include "HL\_sci.h"  
54 #include "HL\_gio.h"  
55  
56 #include <string.h>  
57 #include <stdio.h>  
58 /\* USER CODE END \*/  
59  
60 /\*\* @fn void main(void)  
61 \* @brief Application main function  
62 \* @note This function is empty by default.  
63 \*  
64 \* This function is called after startup.  
65 \* The user can use this function to implement the application.  
66 \*  
67 \*/  
68  
69 /\* USER CODE BEGIN (2) \*/  
70 void sci\_display\_text (sciBASE\_t \*sci, uint8 \*text, uint32 length);  
71 void wait (uint32 time);  
72  
73 #define UART sciREG1  
74  
75 uint32 receive\_data;  
76 uint32 tmp;  
77  
78 /\* USER CODE END \*/  
79  
80 int main(void)  
81 {  
82 /\* USER CODE BEGIN (3) \*/  
  
기본 헤더 설정

Send와 receive를  
프로젝트를 달리  
하였지만 기본적으로  
send 및 receive 동작을  
동시에 할 것 이므로,  
함수나 선언을 가져 와서  
사용함.

Outline HL\_common.h HL\_system.h HL\_sci.h HL\_gio.h string.h stdio.h sci\_display\_text(sciBASE\_t\*, uint8\*, uint32) : void wait(uint32) : void # UART receive\_data : uint32 tmp : uint32 main(void) : int sci\_display\_text(sciBASE\_t\*, uint8\*, uint32) : void wait(uint32) : void

Console No consoles to display at this time.

Problems Advice Memory Allocation Stack Usage 3 items Description Resource Path Location Optimization Advice (3 items)

Writable Smart Insert 77:1:2540

# UART(SCI) – receive



workspace\_v11 - noos\_uart\_receive/source/HL\_sys\_main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

- ▶ eCAP (in RC\_controller\_ecap) [EmbeddedMaster]
- ▶ ecap\_rf\_control
- ▶ external\_gpio\_test
- ▶ nonos\_rtl\_led
- ▶ noos\_etpwm\_control\_with\_uart
- ▶ noos\_uart\_recieve [Active - Debug]

- ▶  Binaries
- ▶  Includes
- ▶  Debug
- ▶  include
- ▶  source
  - ▶  HL\_epc.c
  - ▶  HL\_errata\_SSWF021\_45.c

- ▶ `HL_errata.c`
- ▶ `HL_esm.c`
- ▶ `HL_gio.c`
- ▶ `HL_nmpu.c`
- ▶ `HL_notification.c`

- ▶  [HL\\_sci.c](#)
- ▶  [HL\\_sys\\_core.asm](#)
- ▶  [HL\\_sys\\_dma.c](#)
- ▶  [HL\\_sys\\_intvecs.asm](#)
- ▶  [HL\\_sys\\_link.cmd](#)

- ▶ **HL\_sys\_main.c**
- ▶ **HL\_sys\_mpu.asm**
- ▶ **HL\_sys\_pcr.c**
- ▶ **HL\_sys\_phantom.c**

- ▶ HL\_sys\_pmm.c
- ▶ HL\_sys\_pmu.asm
- ▶ HL\_sys\_startup.c
- ▶ HL\_sys\_vim.c

▶  HL\_system.c

- ▶ targetSettings
- ▶ noos\_uart\_recieve.dil
- ▶ noos\_uart\_recieve.hcg
- ▶ noos\_uart\_send

```
80 int main(void)
81{
82 /* USER CODE BEGIN (3) */
83     char buf[128];
84     unsigned int buf_len;
85
86     sciInit();
87
88     sprintf(buf, "*SCI Init Success!\n\r\0");
89     buf_len = strlen(buf);
90     sci_display_text(UART, (uint8 *)buf, buf_len);
91
92     gioInit();
93
94     sprintf(buf, "*GIO Init Success!\n\r\0");
95     buf_len = strlen(buf);
96     sci_display_text(UART, (uint8 *)buf, buf_len);
97
98     sprintf(buf, "Press Number!\n\r\0");
99     buf_len = strlen(buf);
100    sci_display_text(UART, (uint8 *)buf, buf_len);
101
102
103    for(;)
104    {
105        tmp = sciReceiveByte(UART);
106
107        receive_data = tmp - 48;      // uart 값은 아스키 0
108
109        sprintf(buf, "recv_data = %d\n\r\0", receive_data);
110        buf_len = strlen(buf);
```

## Sprintf 함수를 활용해 Buf에 메시지 내용을 담고

Sci\_display\_text 함수를 활용하여  
Putty 커맨드 화면에 표시.  
[ 3가지 메시지 표시 ]

### **Writables**

Smart Insert

110·1·3269

# UART(SCI) – receive

workspace\_v11 - noos\_uart\_recieve/source/HL\_sys\_main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer

- > eCAP (in RC\_controller\_ecap) [EmbeddedMaster]
- > ecap\_rf\_control
- > external\_gpio\_test
- > nonos\_rti\_led
- > noos\_etpwm\_control\_with\_uart
- > noos\_uart\_recieve [Active - Debug]
  - Binaries
  - Includes
  - Debug
  - include
  - source
    - HL\_epc.c
    - HL\_errata\_SSWF021\_45.c
    - HL\_errata.c
    - HL\_esm.c
    - HL\_gio.c
    - HL\_nmpu.c
    - HL\_notification.c
    - HL\_pinmux.c
    - HL\_sci.c
    - HL\_sys\_core.asm
    - HL\_sys\_dma.c
    - HL\_sys\_intvecs.asm
    - HL\_sys\_link.cmd
    - HL\_sys\_main.c
    - HL\_sys\_mpu.asm
    - HL\_sys\_pcr.c
    - HL\_sys\_phantom.c
    - HL\_sys\_pmm.c
    - HL\_sys\_pmu.asm
    - HL\_sys\_startup.c
    - HL\_sys\_vim.c
    - HL\_system.c
  - targetConfigs
    - noos\_uart\_recieve.dil
    - noos\_uart\_recieve.hcg
  - noos\_uart\_send

```
100    sci_display_text(UART, (uint8 *)buf, buf_len);  
101  
102    for();  
103    {  
104        tmp = sciReceiveByte(UART);  
105  
106        receive_data = tmp - 48; // uart 값은 아스키 이므로 48을 빼서 수자화  
107  
108        sprintf(buf, "recv_data = %d\n\r\0", receive_data);  
109        buf_len = strlen(buf);  
110        sci_display_text(UART, (uint8 *)buf, buf_len);  
111  
112        gioToggleBit(gioPORTA, 4);  
113  
114        sprintf(buf, "you entered %d value! \n\r\0", receive_data);  
115        buf_len = strlen(buf);  
116        sci_display_text(UART, (uint8 *)buf, buf_len);  
117  
118        wait(5000000);  
119    } /* USER CODE END */  
120  
121    return 0;  
122  
123  
124 }  
125  
126  
127 /* USER CODE BEGIN (4) */  
128 void sci_display_text(sciBASE_t *sci, uint8 *text, uint32 length)  
129 {  
130     while(length--)  
131     {
```

sciReceiveByte 함수를 통해 sciREG1의  
데이터 값을 tmp에 저장

Uart 값은 아스키 값  
-> 48을 빼서 숫자 데이터화  
하여야 함.

Receive 한 데이터를  
Display

/dev/ttyUSB0 - PuTTY

```
you entered 2 value!  
recv_data = 7  
you entered 7 value!  
recv_data = 4  
you entered 4 value!  
recv_data = 57  
you entered 57 value!  
recv_data = 0  
you entered 0 value!  
recv_data = 4  
you entered 4 value!  
recv_data = 52  
you entered 52 value!  
recv_data = 54  
you entered 54 value!  
recv_data = 53  
you entered 53 value!  
recv_data = 2  
you entered 2 value!  
recv_data = 1  
you entered 1 value!  
recv_data = 3  
you entered 3 value!
```

Writable

Smart Insert

125:1:3628

# UART(SCI) – receive

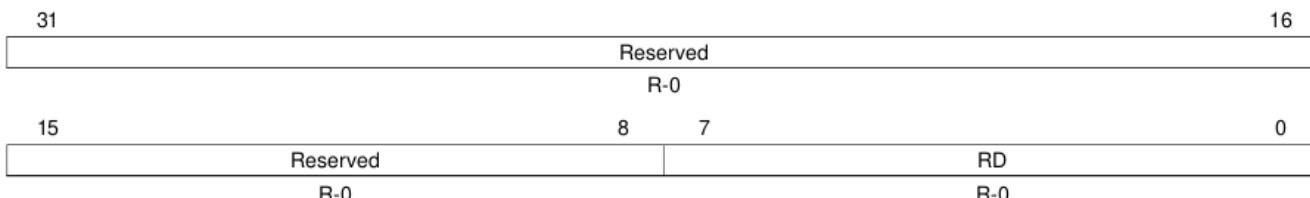
앞 페이지의 sciReceiveByte

```
419 uint32 sciReceiveByte(sciBASE_t *sci)
420 {
421 /* USER CODE BEGIN (16) */
422 /* USER CODE END */
423
424 /*SAFETYMCUSW 28 D MR:NA <APPROVED> "Potentially infinite loop found - Hardware Status check for execution seq
425 while ((sci->FLR & (uint32)SCI_RX_INT) == 0U)
426 {
427 /* Wait */
428
429 return (sci->RD & (uint32)0x000000FFU);
430 }
```

## 30.7.12.2 Receiver Data Buffer (SCIRD)

This register provides a location for the receiver data. [Figure 30-20](#) and [Table 30-19](#) illustrate this register.

Figure 30-20. Receiver Data Buffer (SCIRD) [offset = 34h]



LEGEND: R = Read only; -n = value after reset

Table 30-19. Receiver Data Buffer (SCIRD) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reads return 0. Writes have no effect.
7-0	RD	0-FFh	Receiver data. When a frame has been completely received, the data in the frame is transferred from the receiver shift register SCIRXSHF to this register. As this transfer occurs, the RXRDY flag (SCIFLR[9]) is set and a receive interrupt is generated if SET RX INT bit (SCISETINT[9]) is set. <b>Note:</b> When the data is read from SCIRD, the RXRDY flag (SCIFLR[9]) is automatically cleared.

**NOTE:** When the SCI receives data that is fewer than eight bits in length, it loads the data into this register in a left-justified format padded with trailing zeros. Therefore, the user software should perform a logical shift on the data by the correct number of positions to make it right justified.