EDDI
Electronic Design
Development Institute

에디로봇아카데미

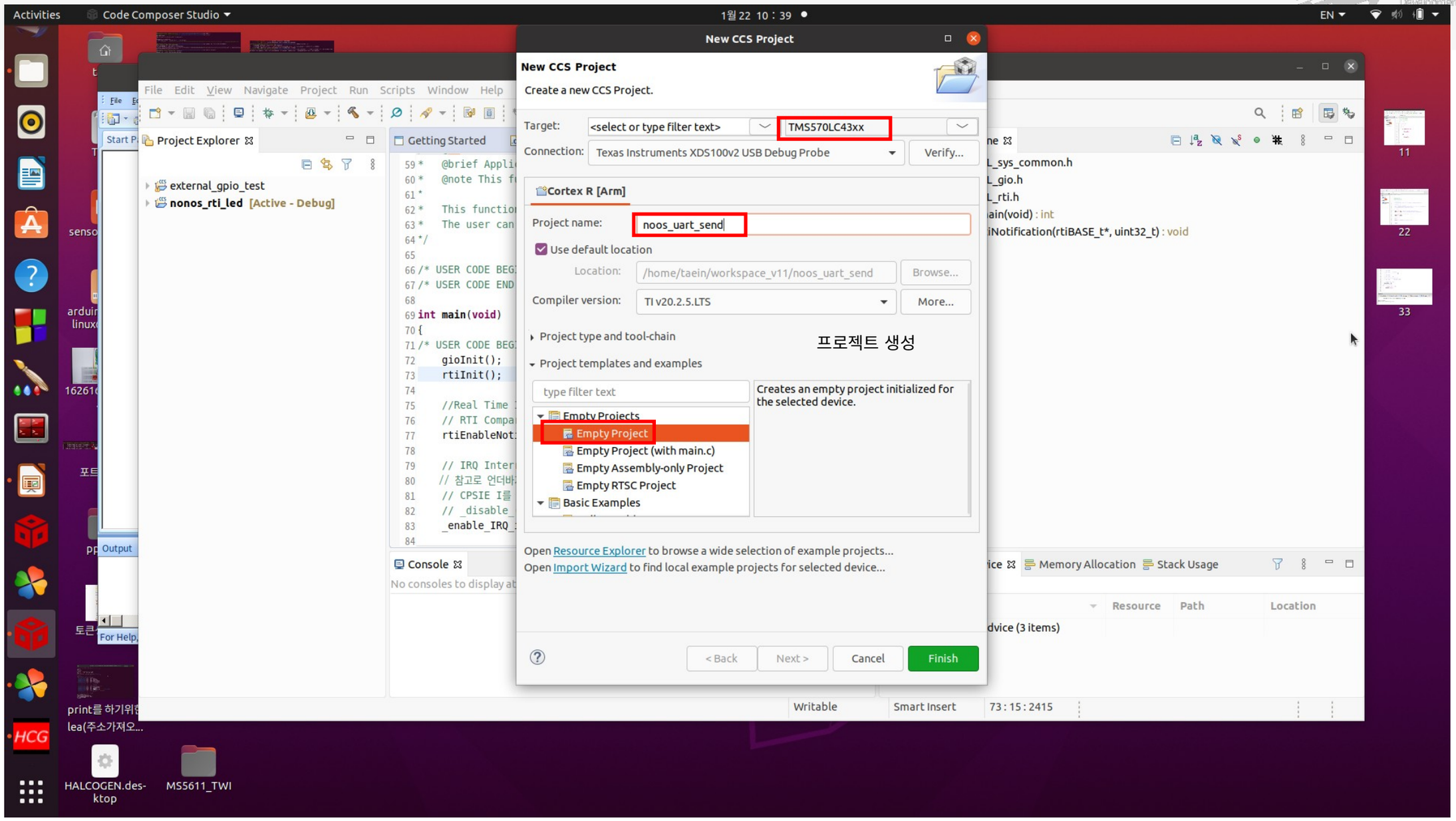
임베디드 마스터 Lv2 과정
[TMS 570_UART(SCI)]

제 1기

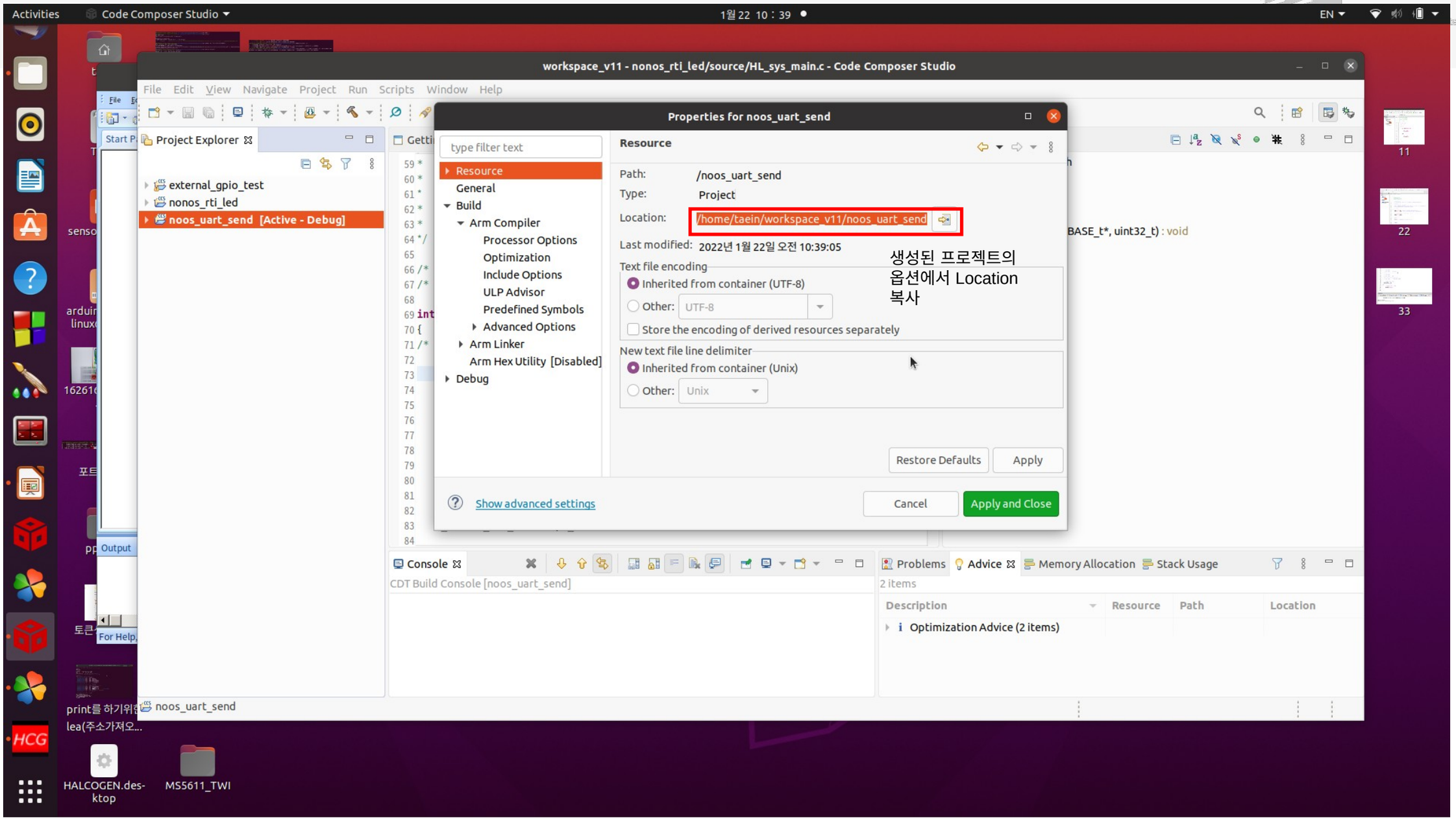
2022. 02. 18

박태인

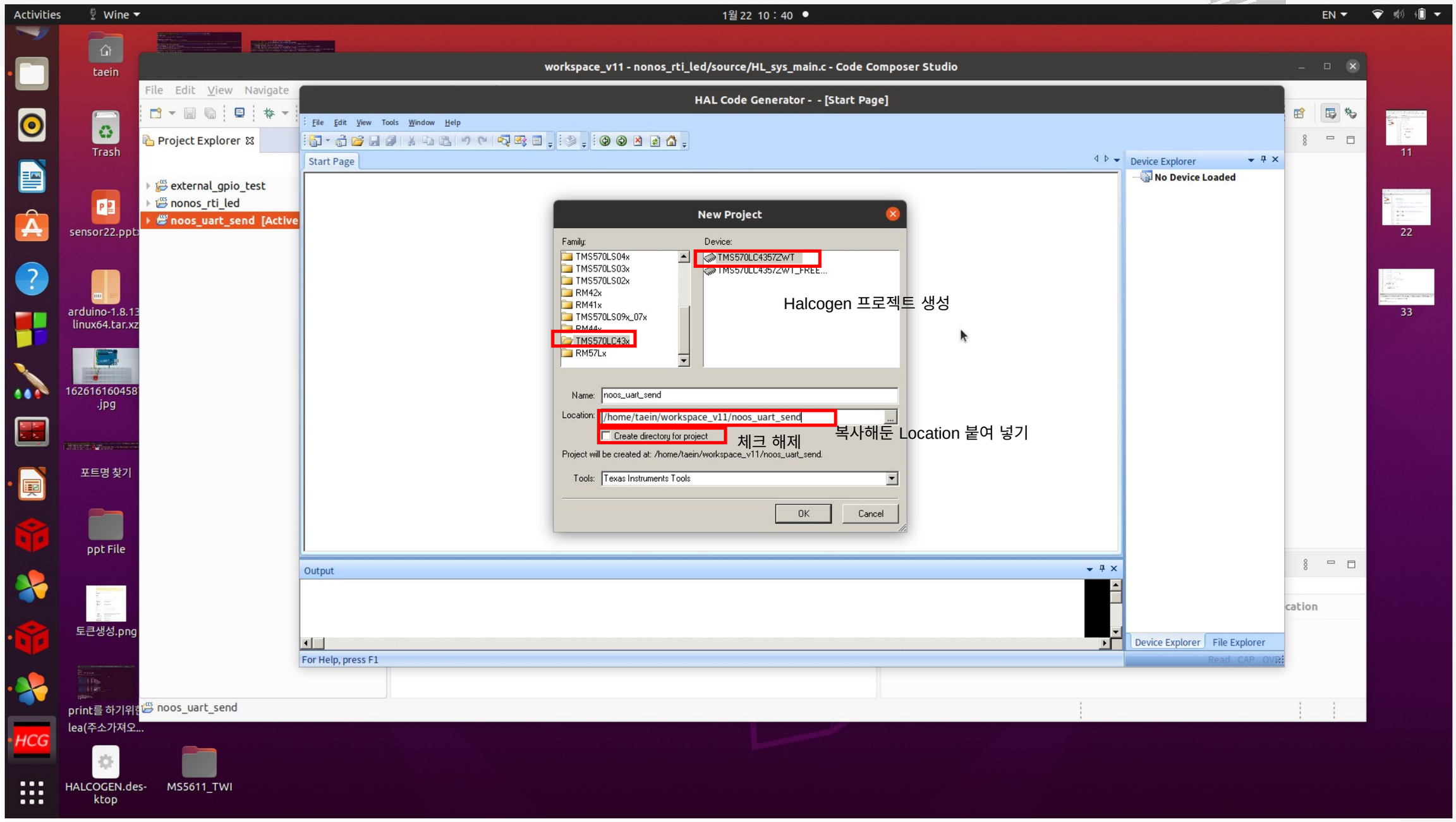
UART(SCI) – send



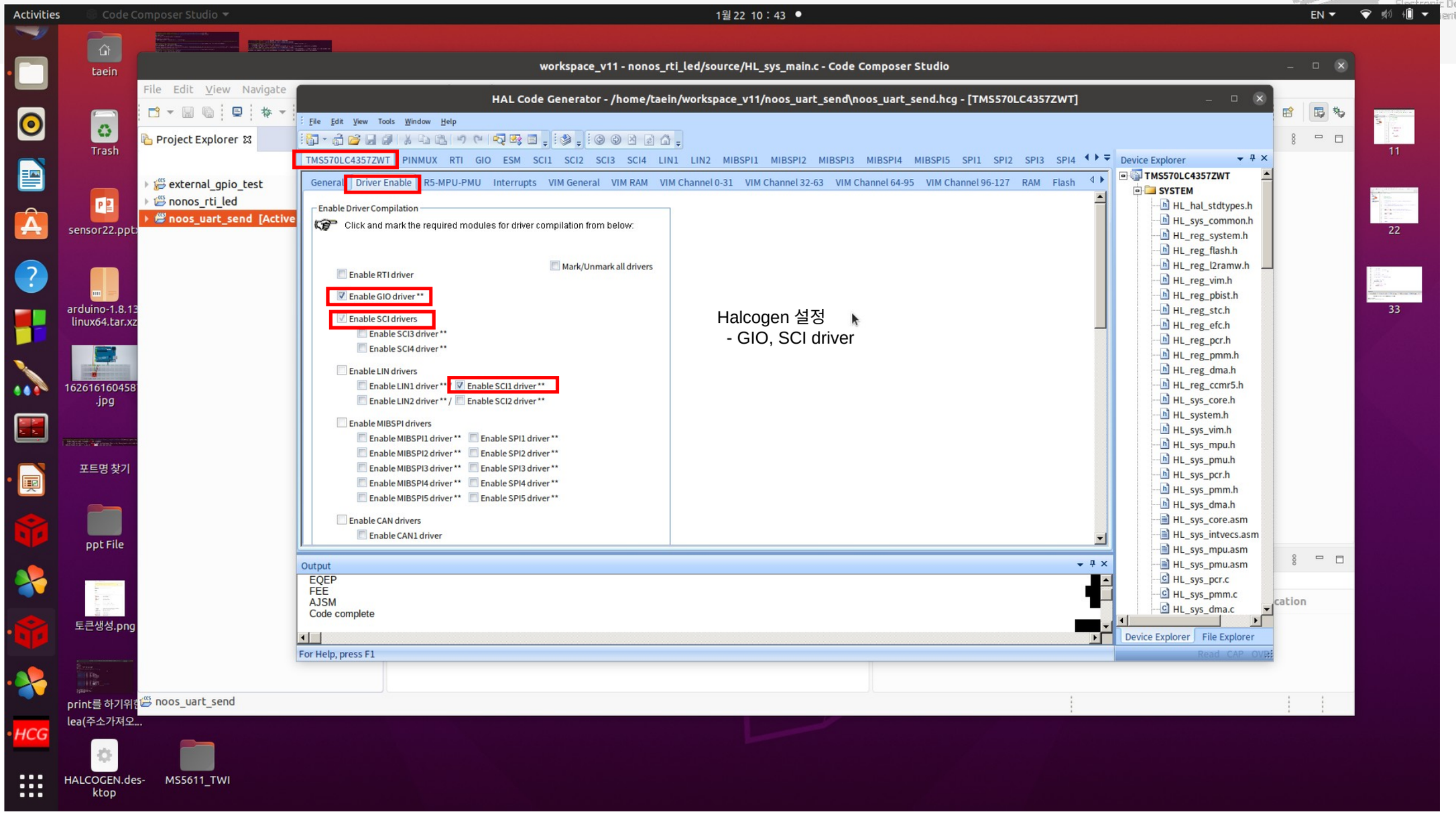
UART(SCI) - send



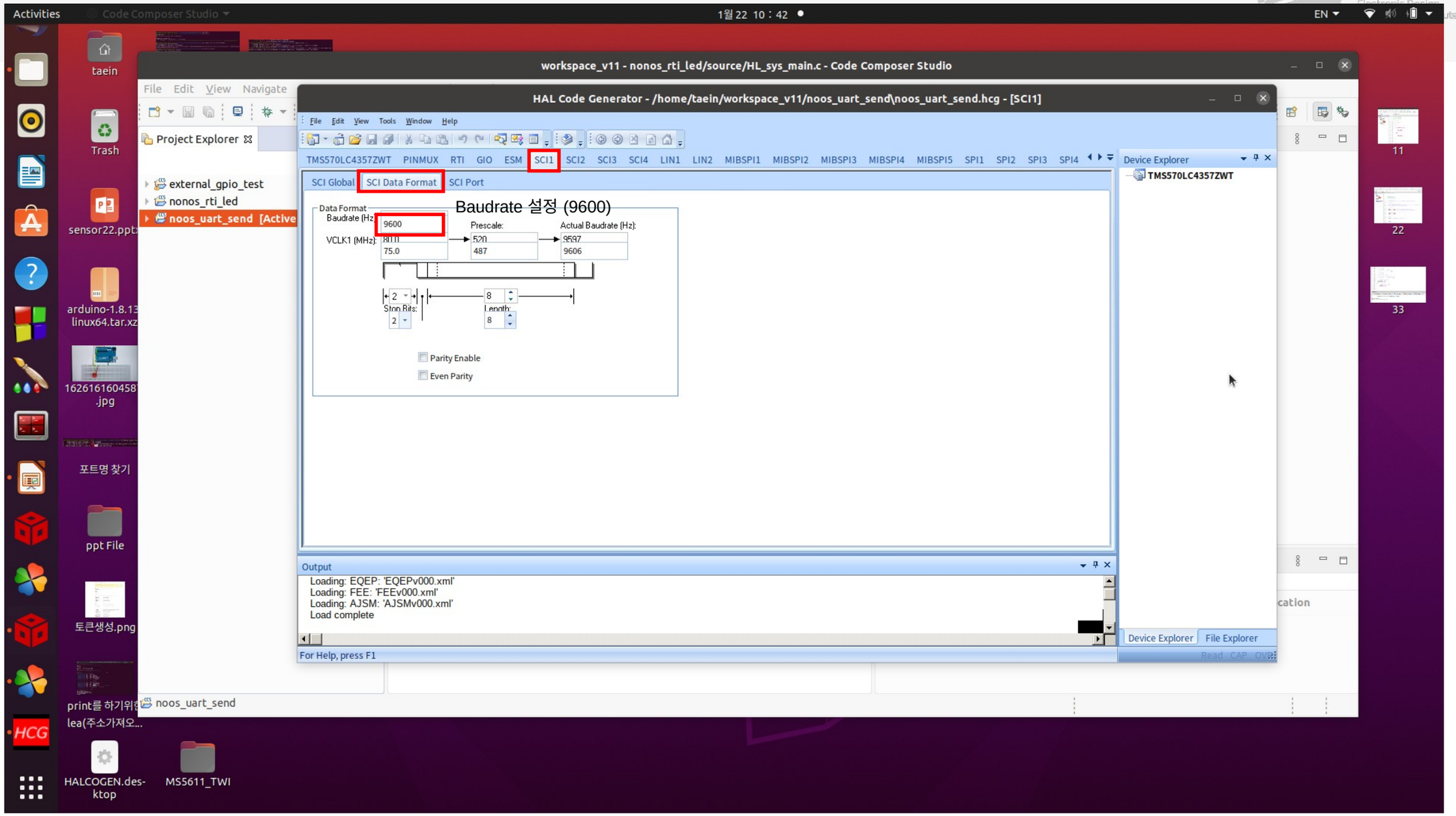
UART(SCI) - send



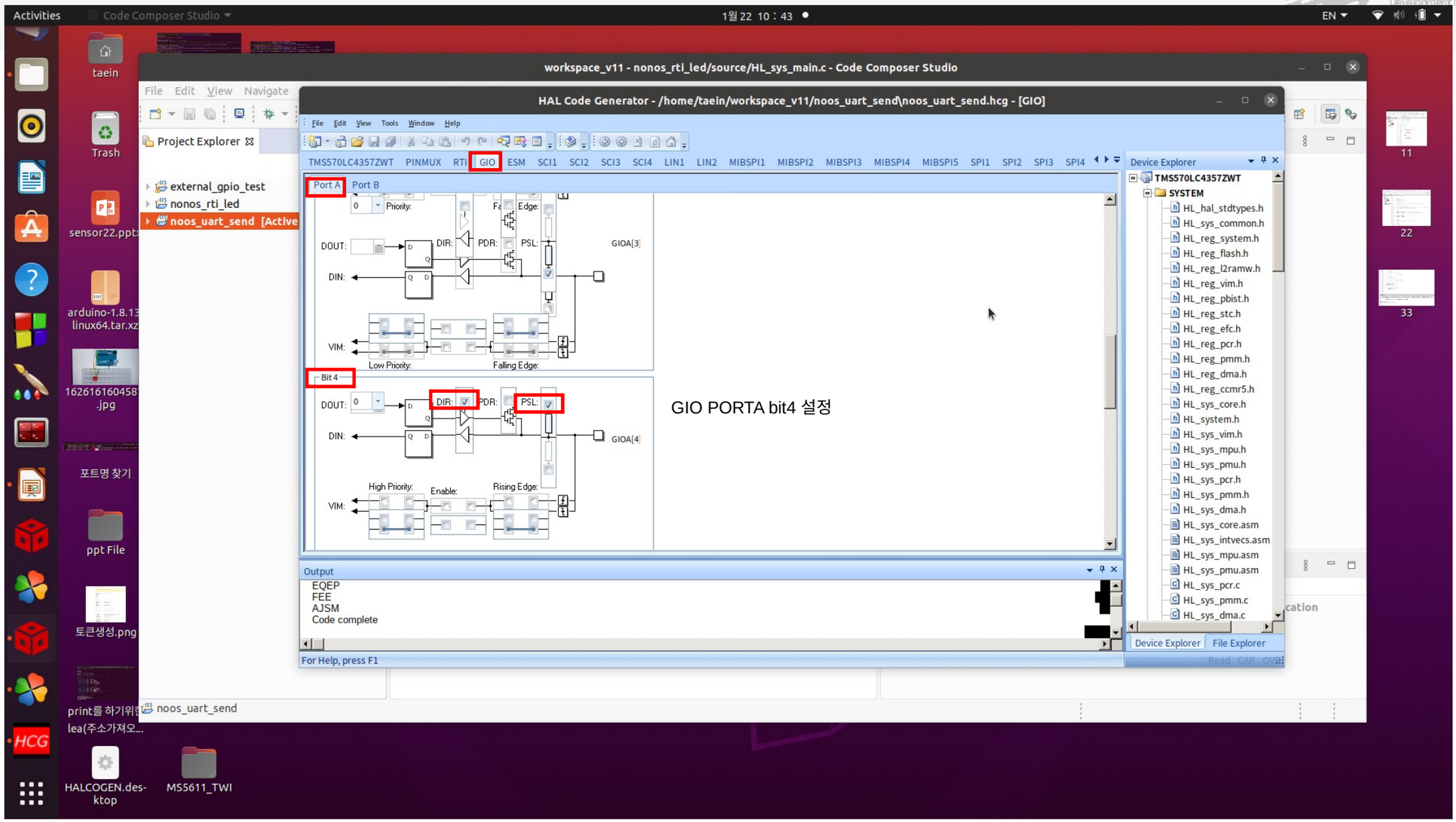
UART(SCI) - send



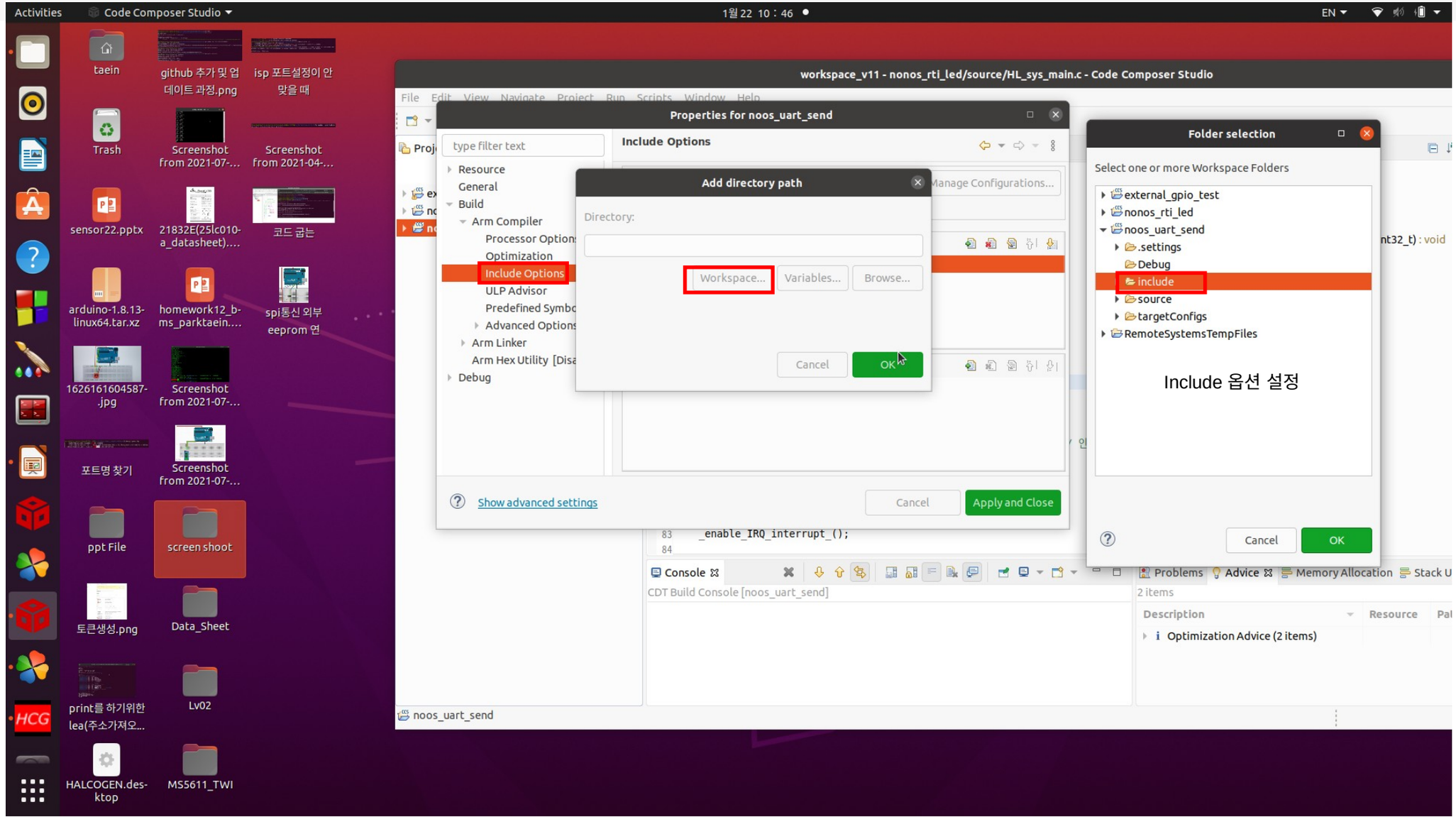
UART(SCI) - send



UART(SCI) - send

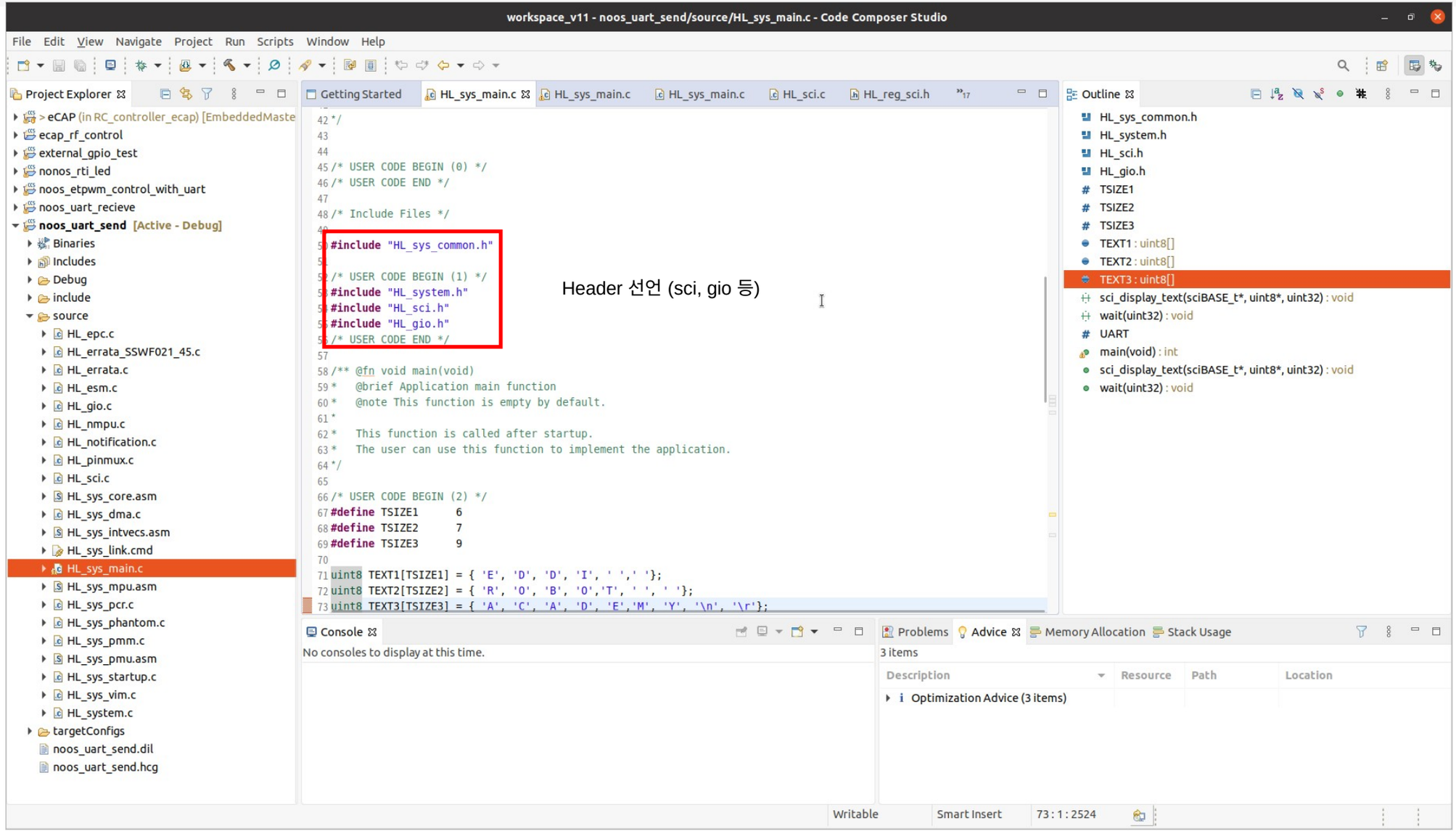


UART(SCI) - send

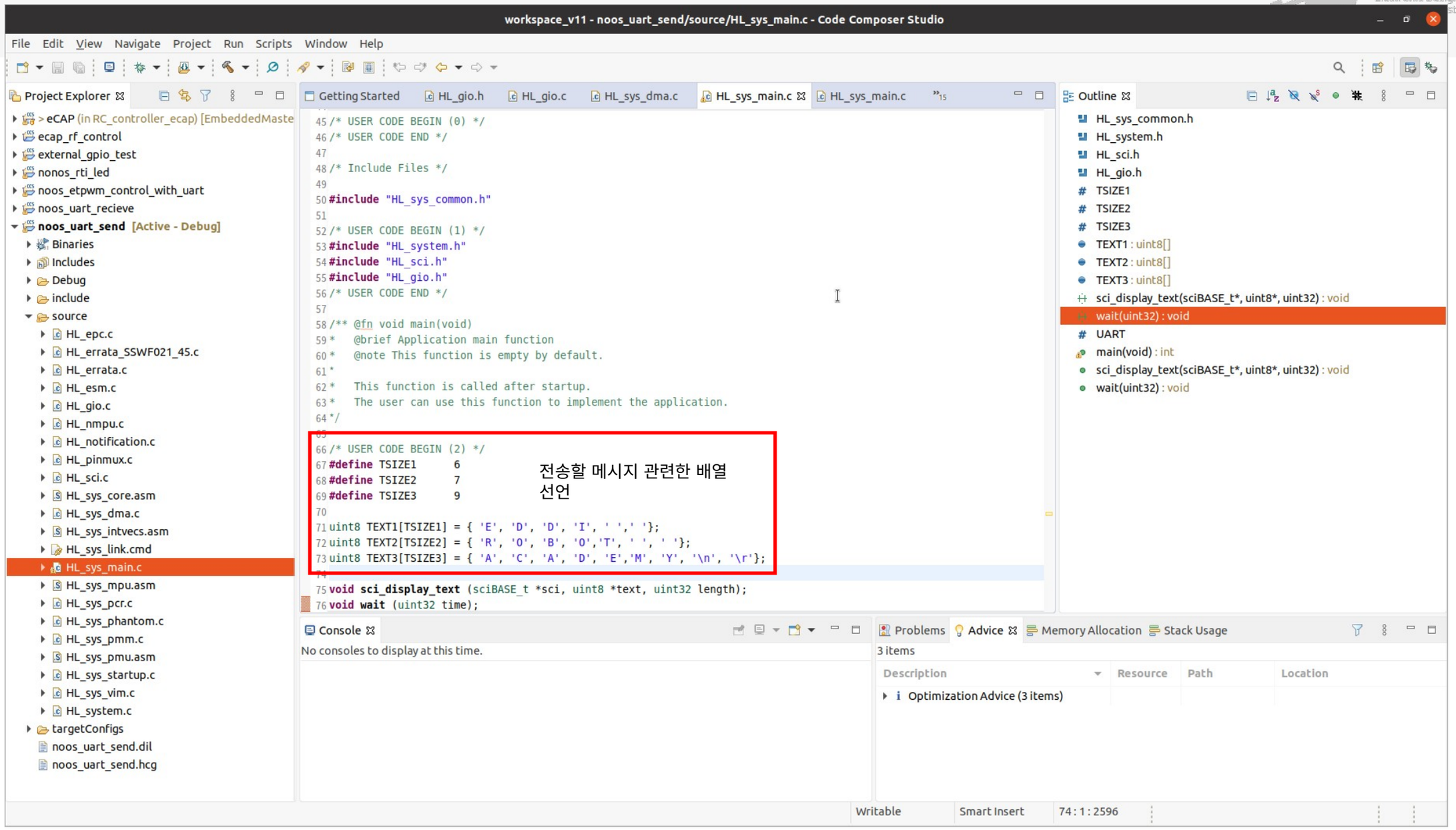


UART(SCI) - send

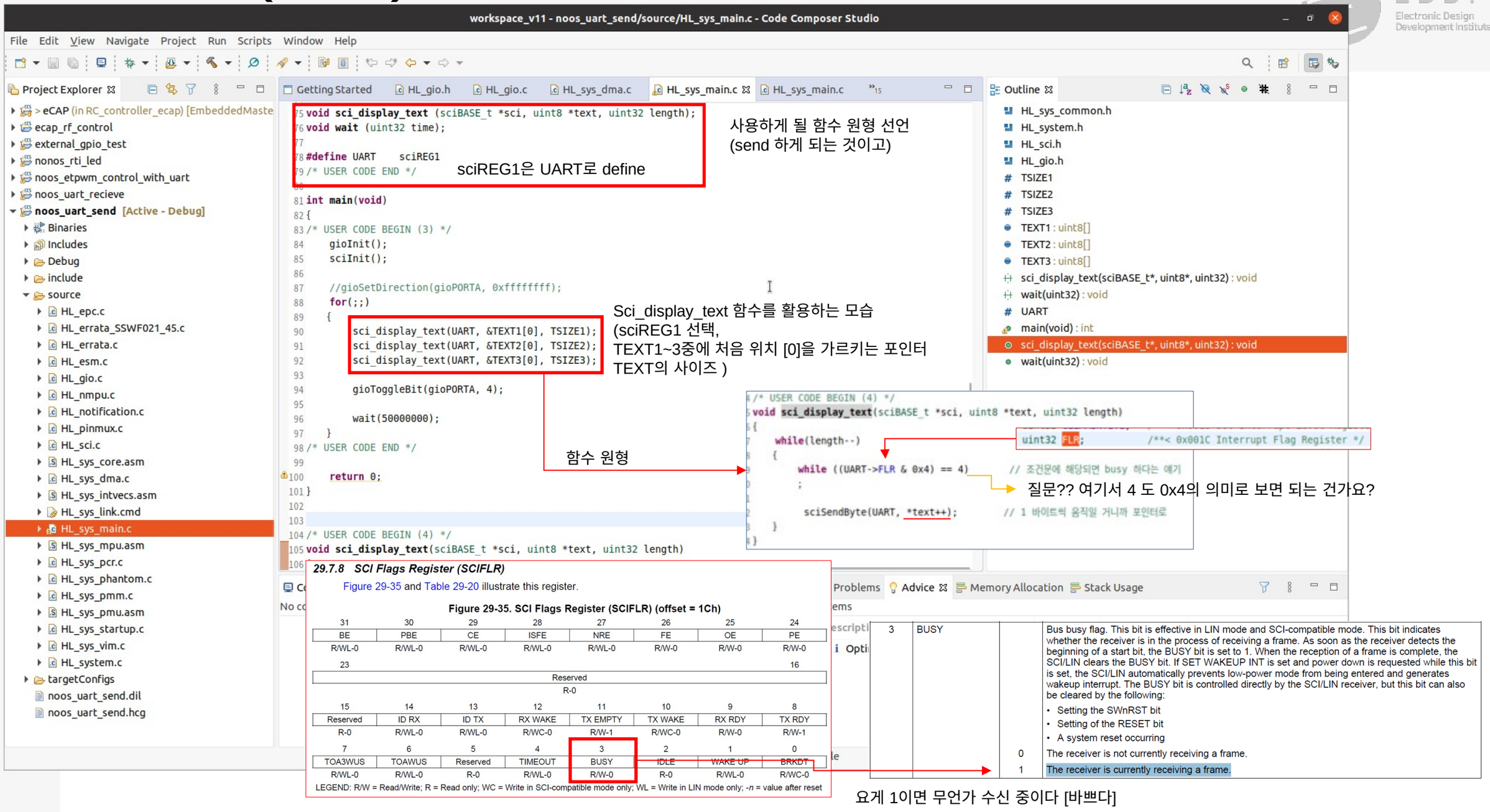
Code 분석



UART(SCI) - send



UART(SCI) - send




```
void sci_display_text (sciBASE_t *sci, uint8 *text, uint32 length);
void wait (uint32 time);

#define UART    sciREG1
/* USER CODE END */

int main(void)
{
/* USER CODE BEGIN (3) */
    gioInit();
    sciInit();

    //gioSetDirection(gioPORTA, 0xffffffff);
    for(;;)
    {
        sci_display_text(UART, &TEXT1[0], TSIZE1);
        sci_display_text(UART, &TEXT2[0], TSIZE2);
        sci_display_text(UART, &TEXT3[0], TSIZE3);

        gioToggleBit(gioPORTA, 4);

        wait(50000000);
    }
/* USER CODE END */

    return 0;
}

/* USER CODE BEGIN (4) */
void sci_display_text(sciBASE_t *sci, uint8 *text, uint32 length)
```

sciREG1은 UART로 define

사용하게 될 함수 원형 선언
(send 하게 되는 것이고)

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

함수 원형

```
/* 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)

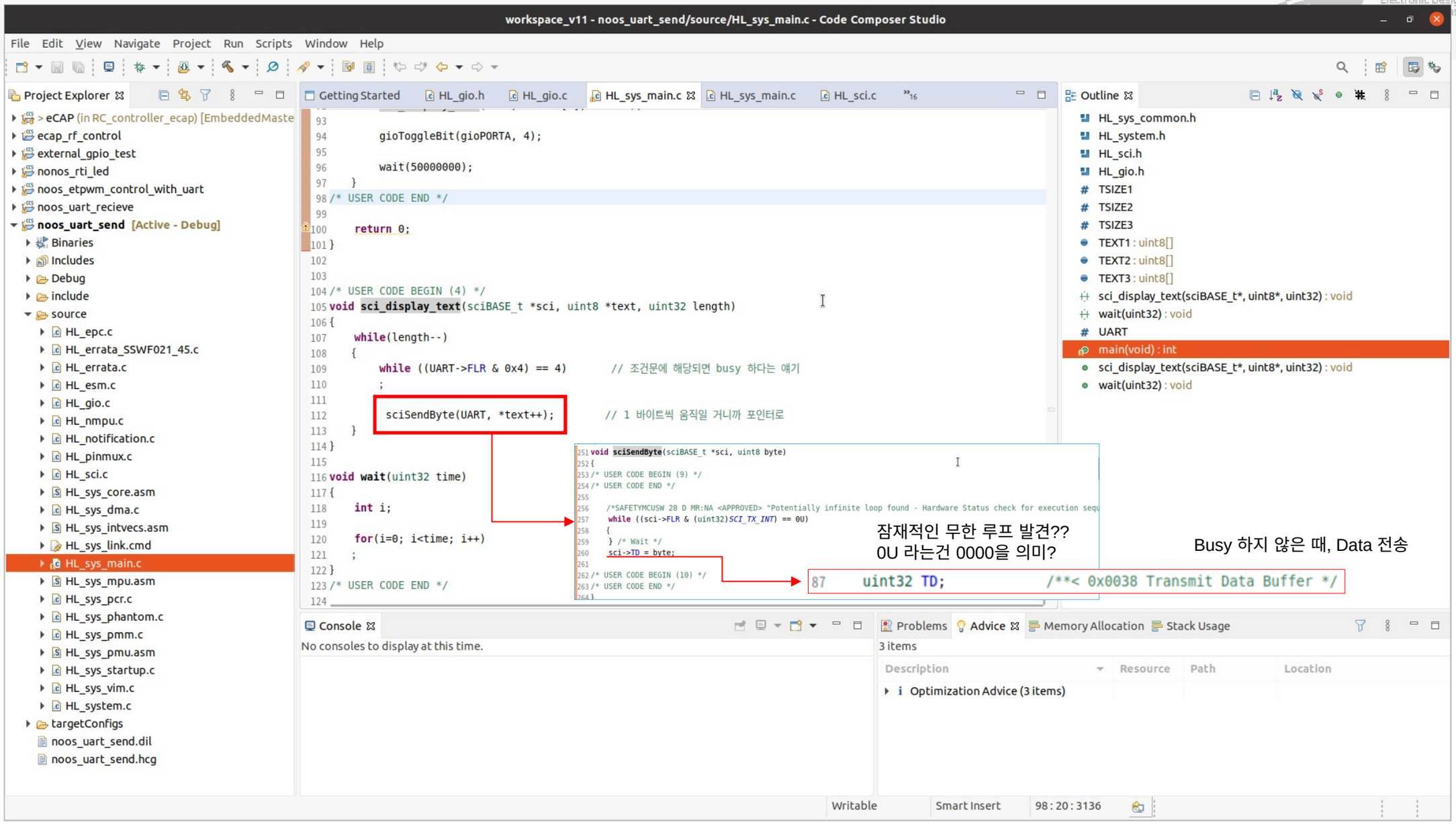
31	30	29	28	27	26	25	24
BE	PBE	CE	ISFE	NRE	FE	OE	PE
R/WL-0	R/WL-0	R/WL-0	R/WL-0	R/WL-0	R/W-0	R/W-0	R/W-0
23							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved	ID RX	ID TX	RX WAKE	TX EMPTY	TX WAKE	RX RDY	TX RDY
R-0	R/WL-0	R/WL-0	R/WC-0	R/W-1	R/WC-0	R/W-0	R/W-1
7	6	5	4	3	2	1	0
TOA3WUS	TOAWUS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

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 wakeup 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.

요게 1이면 무언가 수신 중이다 [바쁘다]

UART(SCI) - send



UART(SCI) - send

SciInit 분석

EDDI
Electronic Design Development Institute

```
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]

31	16
Reserved	
R-0	

15	1	0
Reserved		RESET
R-0		R/WP-0

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) Fied Descriptions

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

UART(SCI) - send

SciInit 분석

EDDI Electronic Design Development Institute

```
/** - Disable all interrupts */
sciREG1->CLEARINT    = 0xFFFFFFFFU;
sciREG1->CLEARINTLVL = 0xFFFFFFFFU;
```

```
uint32 CLEARINT;      /**< 0x0010 Clear Interrupt Enable Register */
uint32 CLEARINTLVL;   /**< 0x0018 Set Interrupt Level Register */
```

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

4	CLR TIMEOUT INT		Clear timeout interrupt. This bit is effective in LIN mode only. This bit disables the timeout (LIN bus idle) interrupt when set.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read:* The interrupt is enabled. *Write:* 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		Clear timeout interrupt. This bit is effective in LIN mode only.
		0	*Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The interrupt level is mapped to the INT1 line. *Write:* 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

SciInit 분석

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

```
/** - 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 */
```

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

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

UART(SCI) - send

SciInit 분석

```
uint32 BRS;
uint32 FORMAT;
```

```
/**< 0x002C Baud Rate Selection Register */
/**< 0x0028 Format Control Register */
```

```
/** - 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 Section 29.3.1.4.1 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 Section 29.3.1.4.1 for more details.
23-0	PRESCALER P	0-FF FFFFh	These bits are used to select a baud rate for the SCI/LIN module. These bits are effective in LIN mode and SCI compatibility. 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]. **NOTE: In LIN mode, ONLY the asynchronous mode and baud rate values are used.** The baud rate can be calculated using the following formulas: $\text{Asynchronous baud value} = \left(\frac{\text{VCLK Frequency}}{16\left(P + 1 + \frac{M}{16}\right)}\right)$ (52) $\text{Isosynchronous baud value} = \left(\frac{\text{VCLK Frequency}}{P + 1}\right)$ (53) For P = 0, $\text{Asynchronous baud value} = \left(\frac{\text{VCLK Frequency}}{32}\right)$ (54) $\text{Isosynchronous baud value} = \left(\frac{\text{VCLK Frequency}}{2}\right)$ (55) Table 29-25 contains comparative baud values for different P values, with VCLK = 50 MHz, for asynchronous mode.

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

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

(1) VCLK = 50 MHz
(2) Values are in decimal except for column 2.

30.7.10 SCI Format Control Register (SCIFORMAT)

Figure 30-17 and Table 30-15 illustrate this register.

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

31	16
Reserved	
R-0	

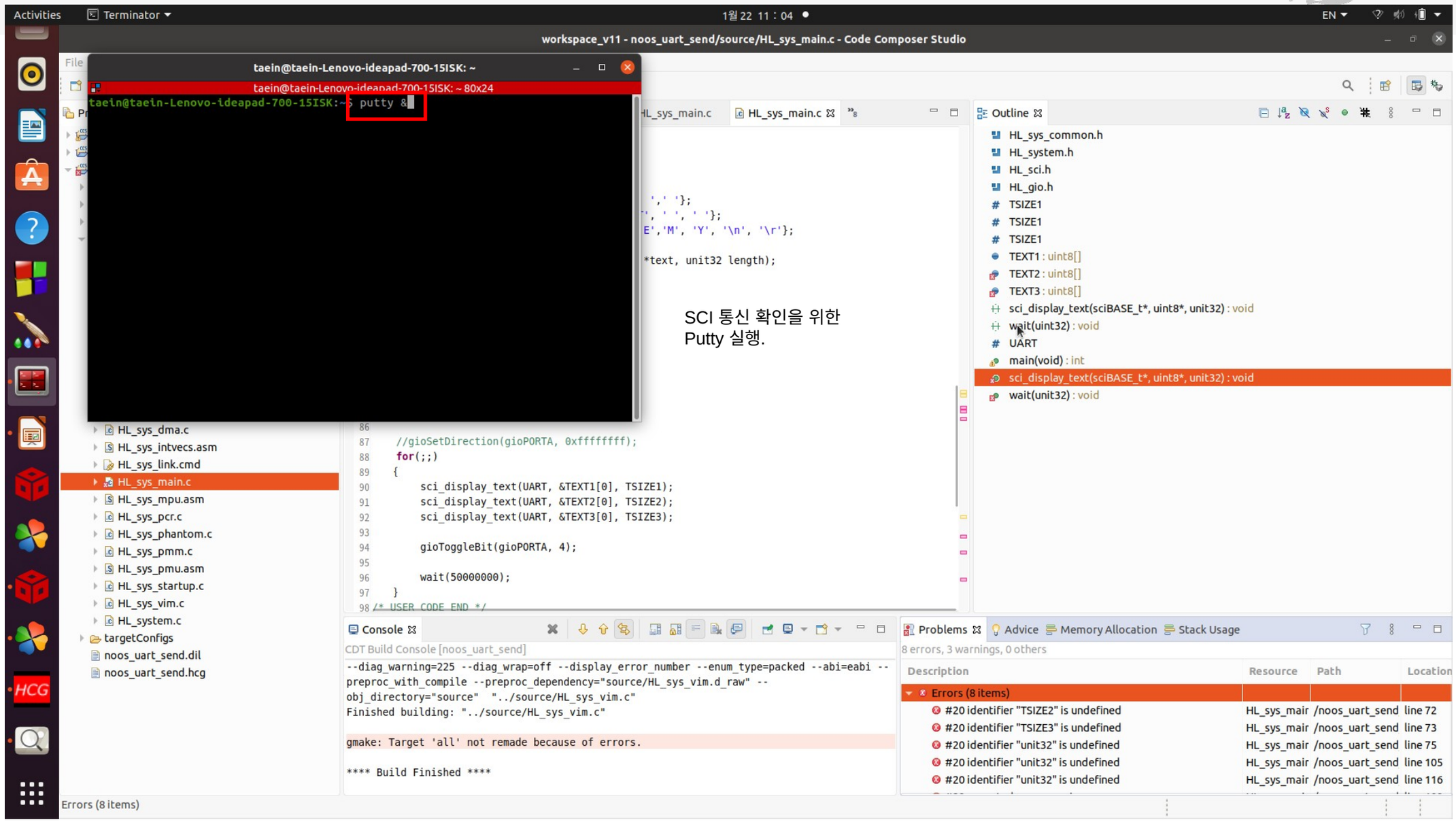
15	3	2	0
Reserved		CHAR	
R-0		R/W-0	

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

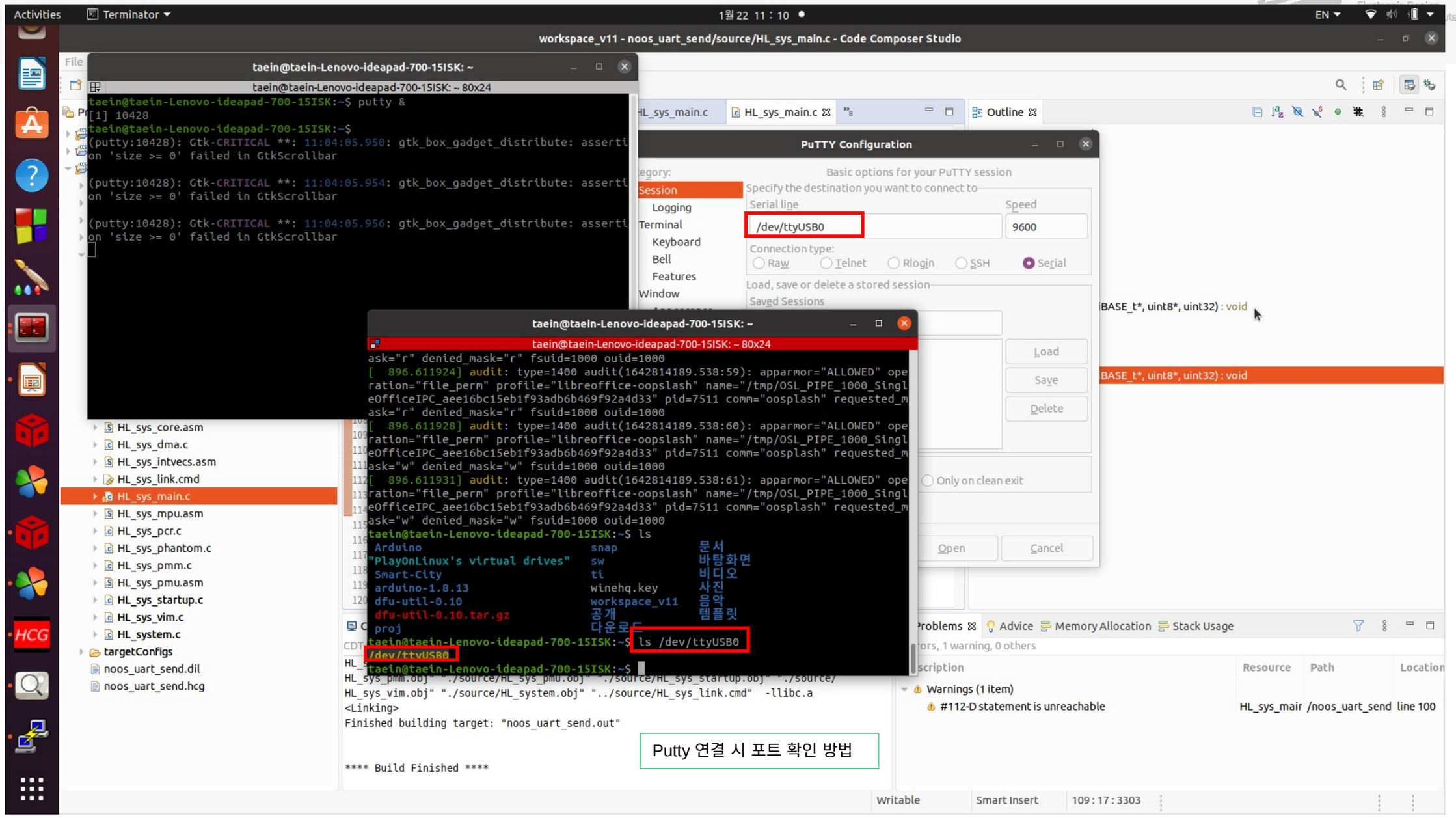
Table 30-15. SCI Format Control Register (SCIFORMAT) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reads return 0. Writes have no effect.
2-0	CHAR		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.**
		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.

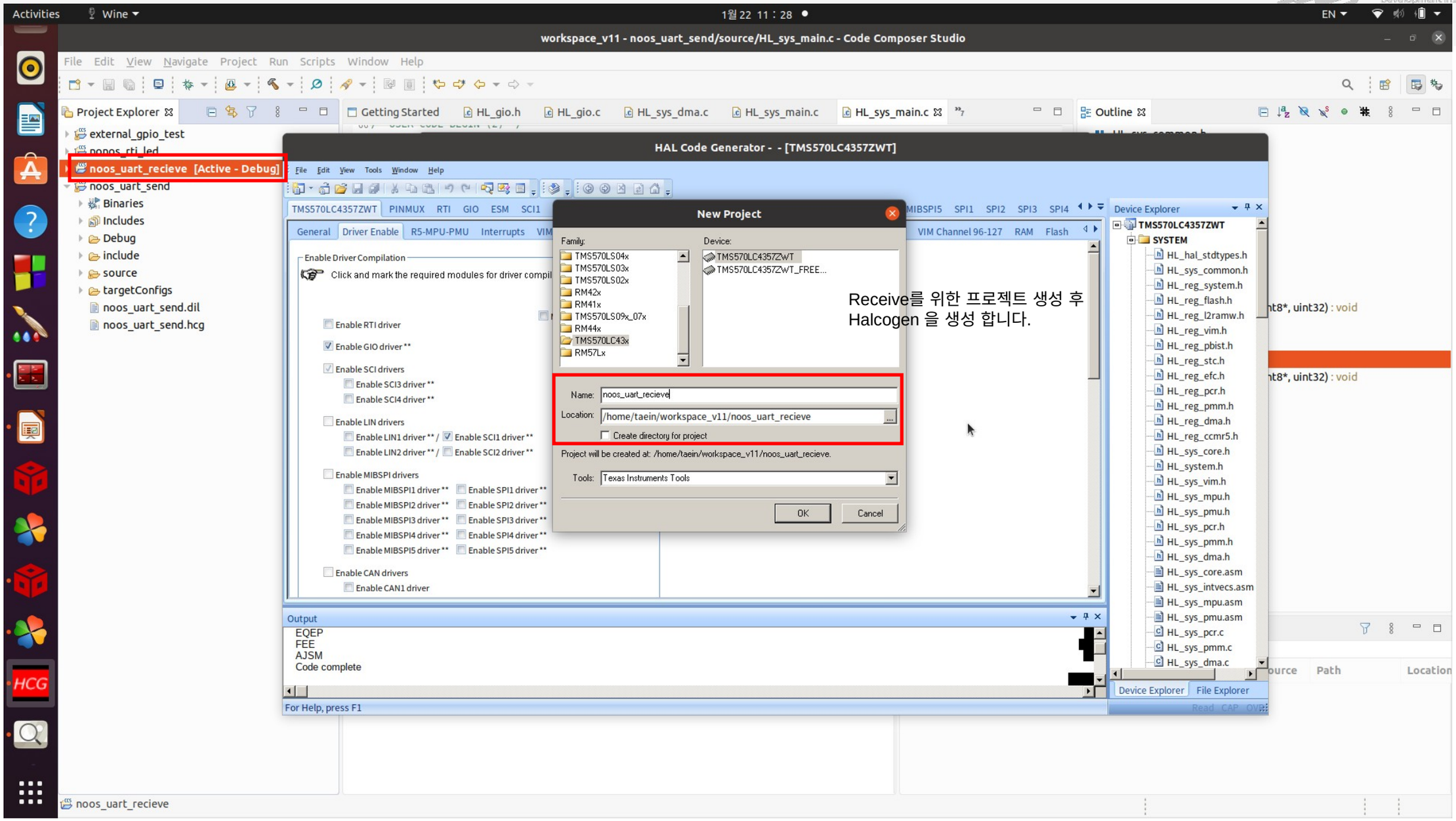
UART(SCI) - send



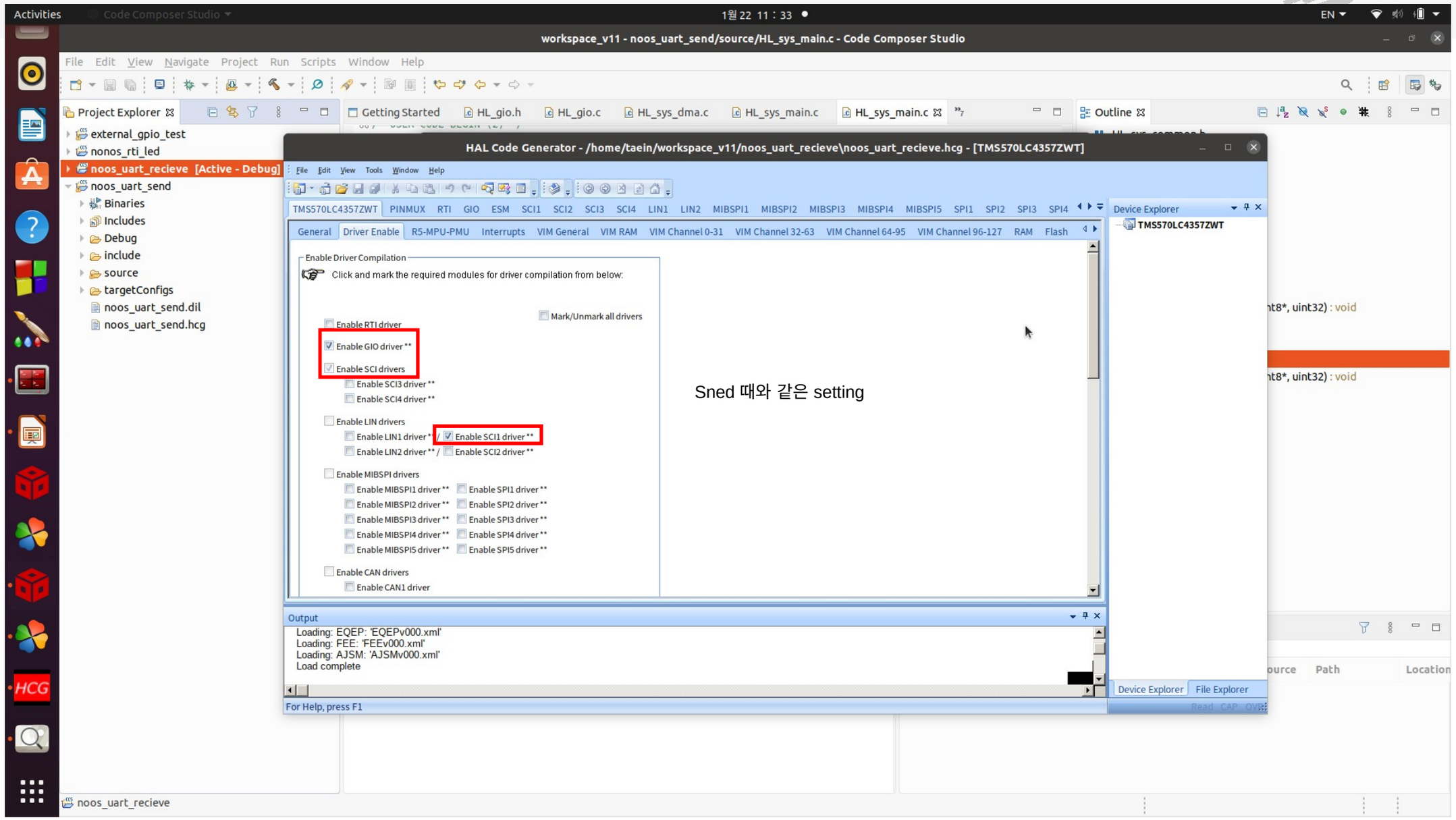
UART(SCI) - send



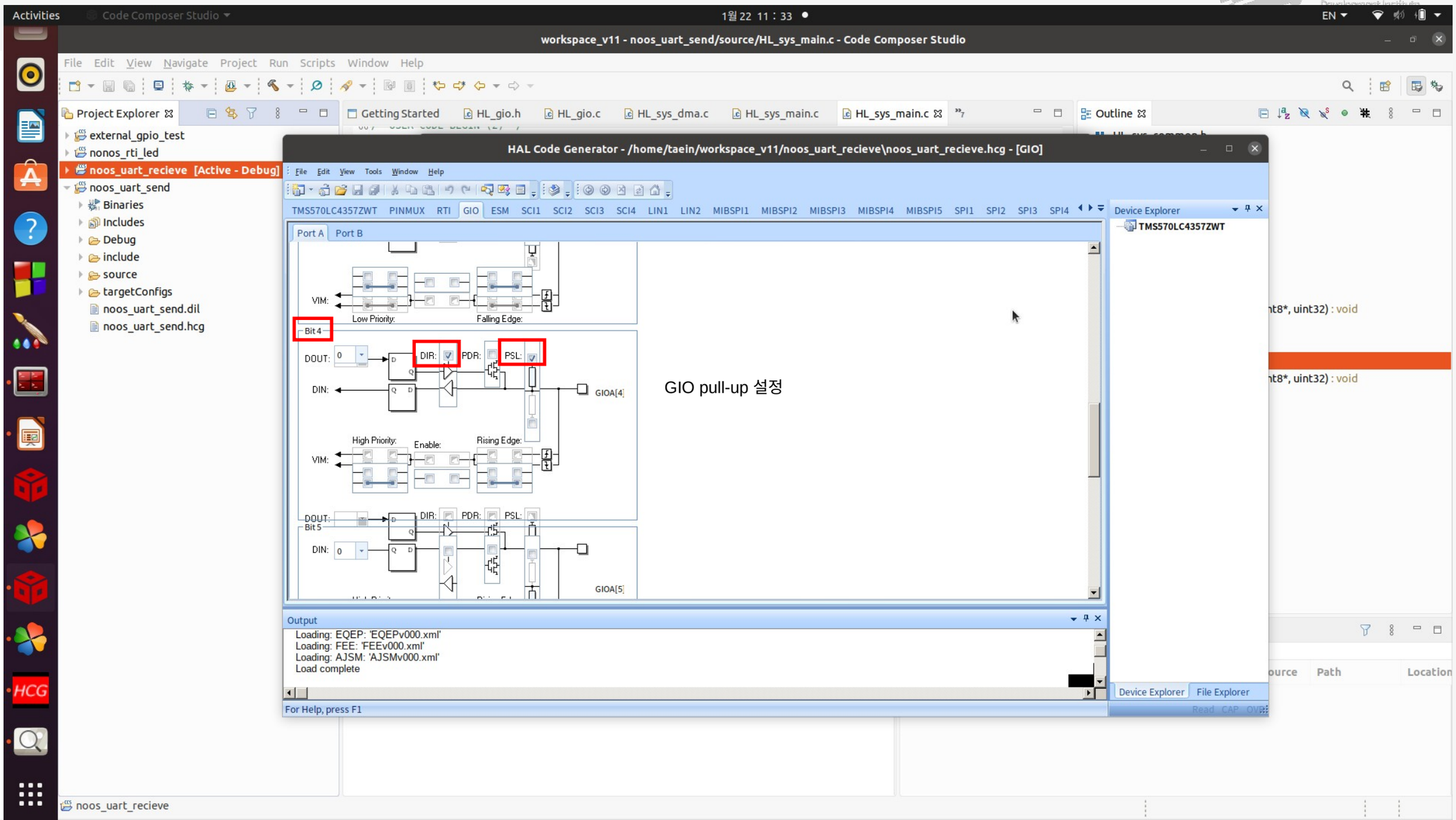
UART(SCI) – receive



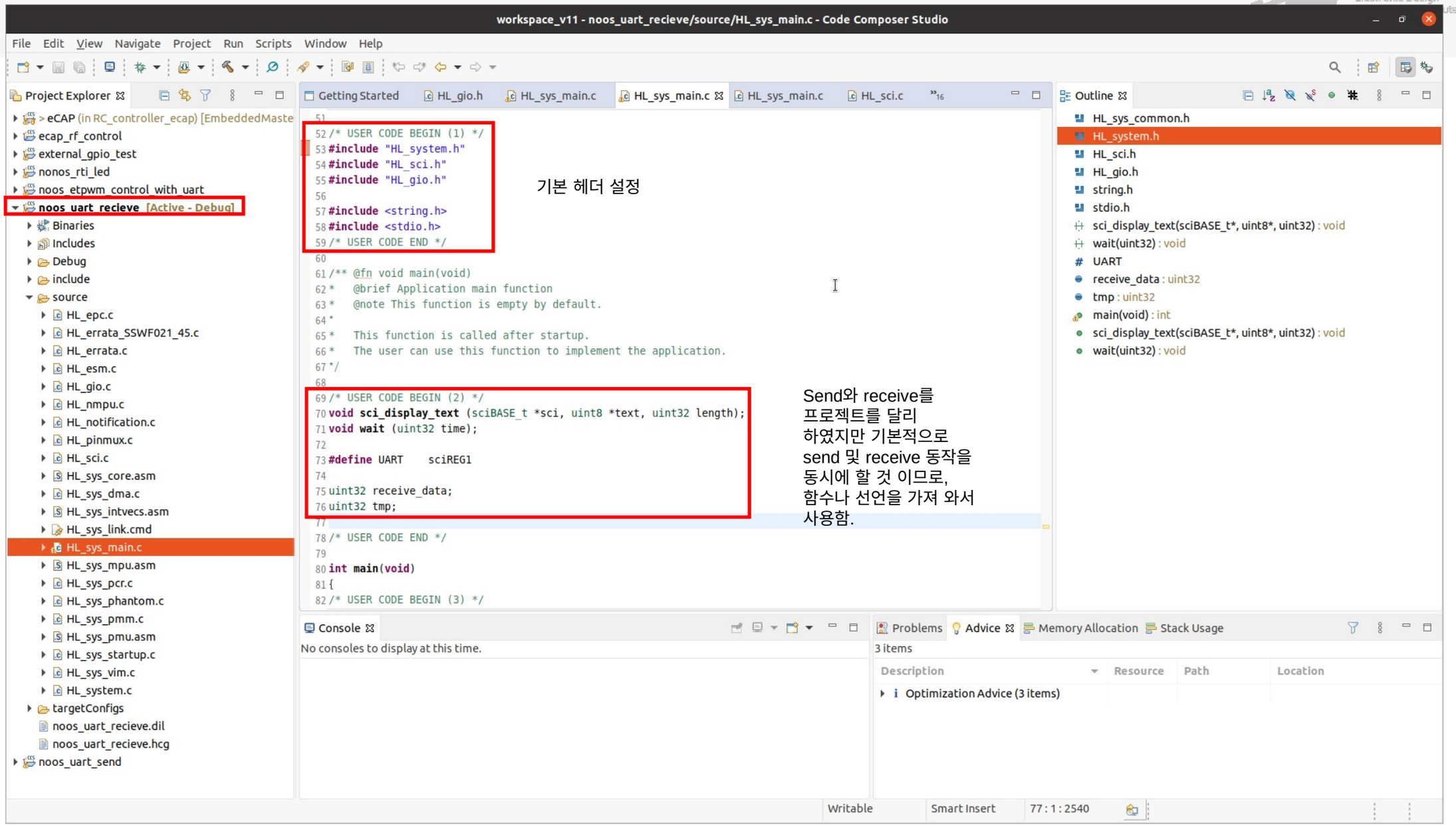
UART(SCI) – receive



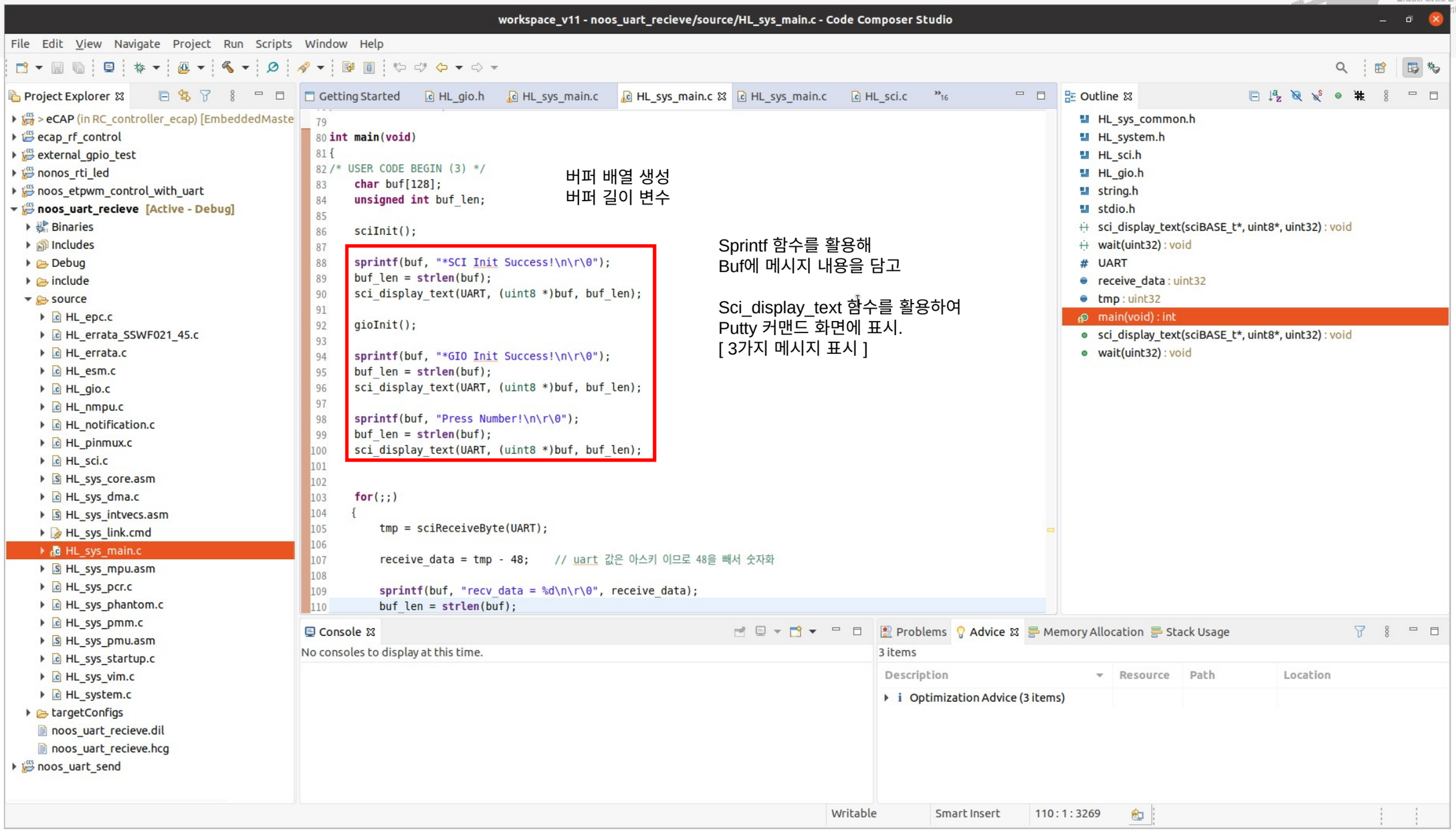
UART(SCI) – receive



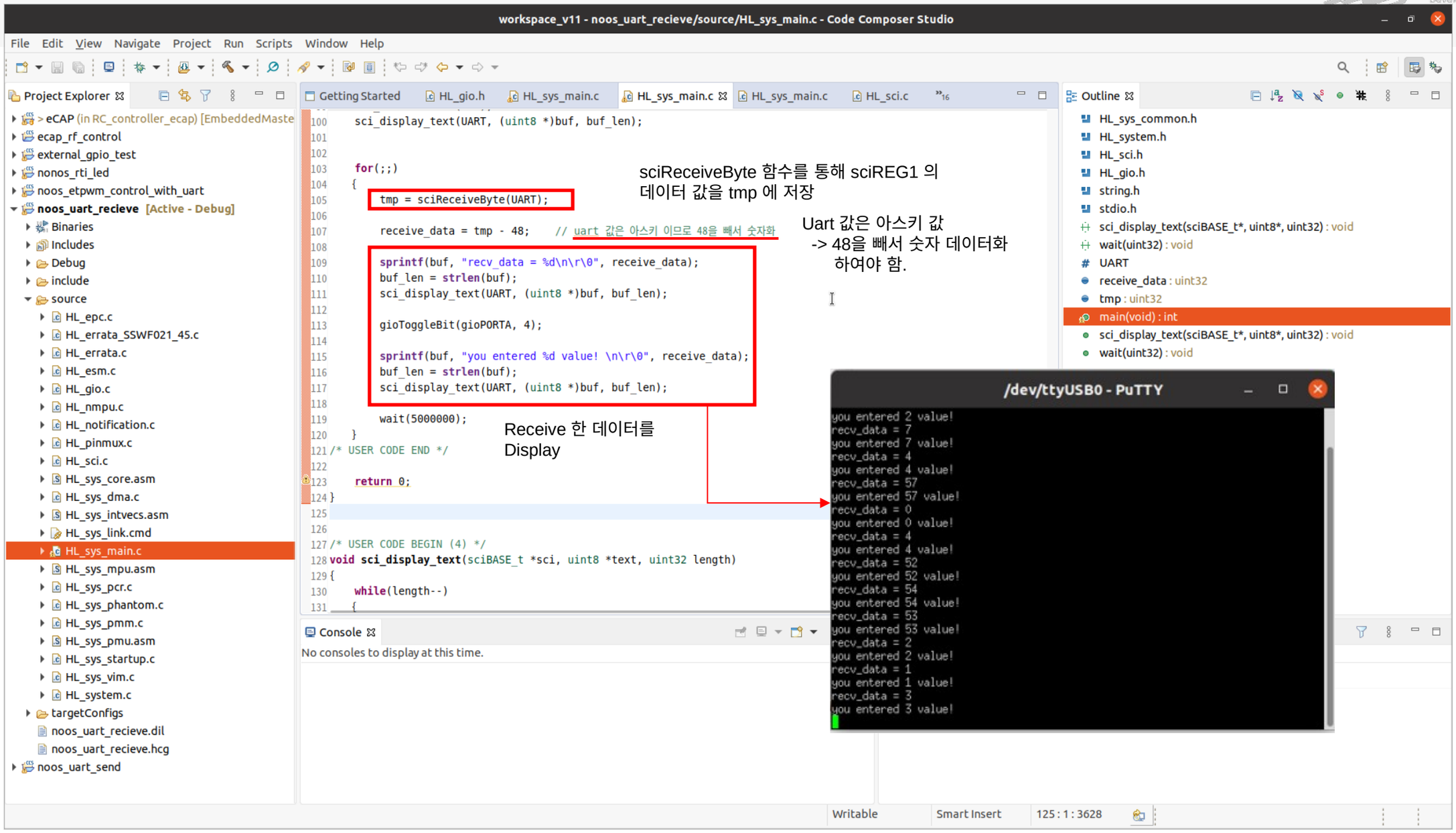
UART(SCI) – receive



UART(SCI) – receive



UART(SCI) – receive



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]

31								16
Reserved								
R-0								

15							8	7							0
Reserved								RD							
R-0								R-0							

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. **Note: 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.