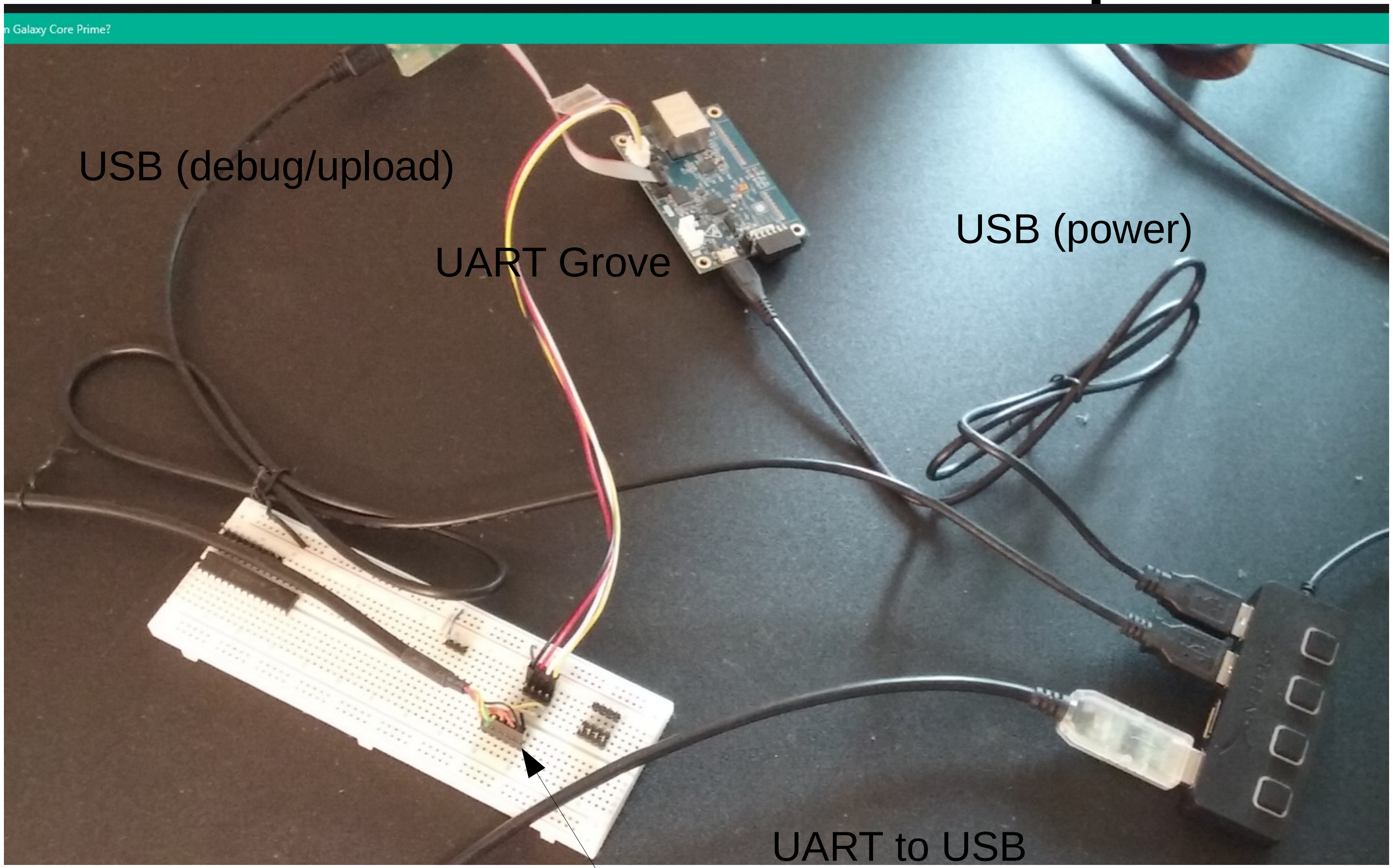
S5D9 UART Bus Example (Driver Version)

By
Michael Li
(2/2/2018)
https://www.miketechuniverse.com

E2 Studio 5.4.0.023
SSP 1.3.0

Hardware Setup



UART to USB
FTDI cable
(From Adafruit)

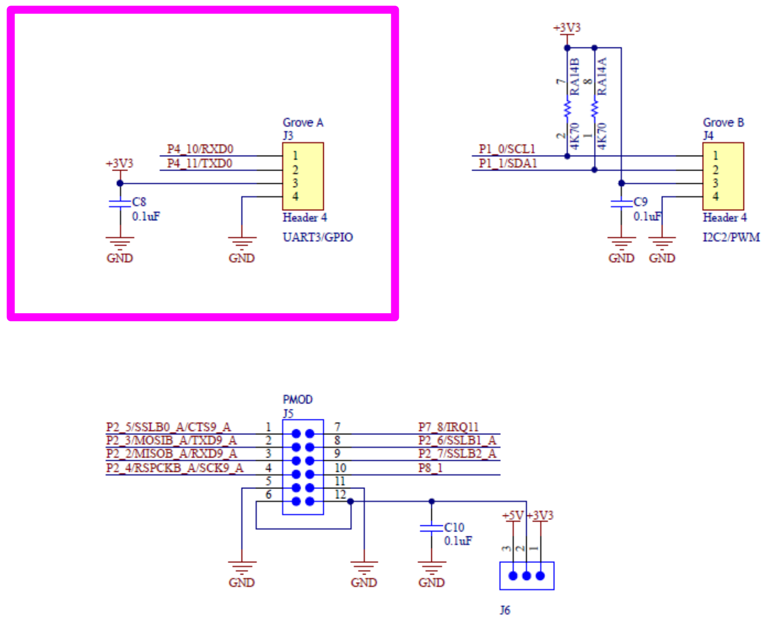
Color code (see next page)

Hardware Connection

Connections

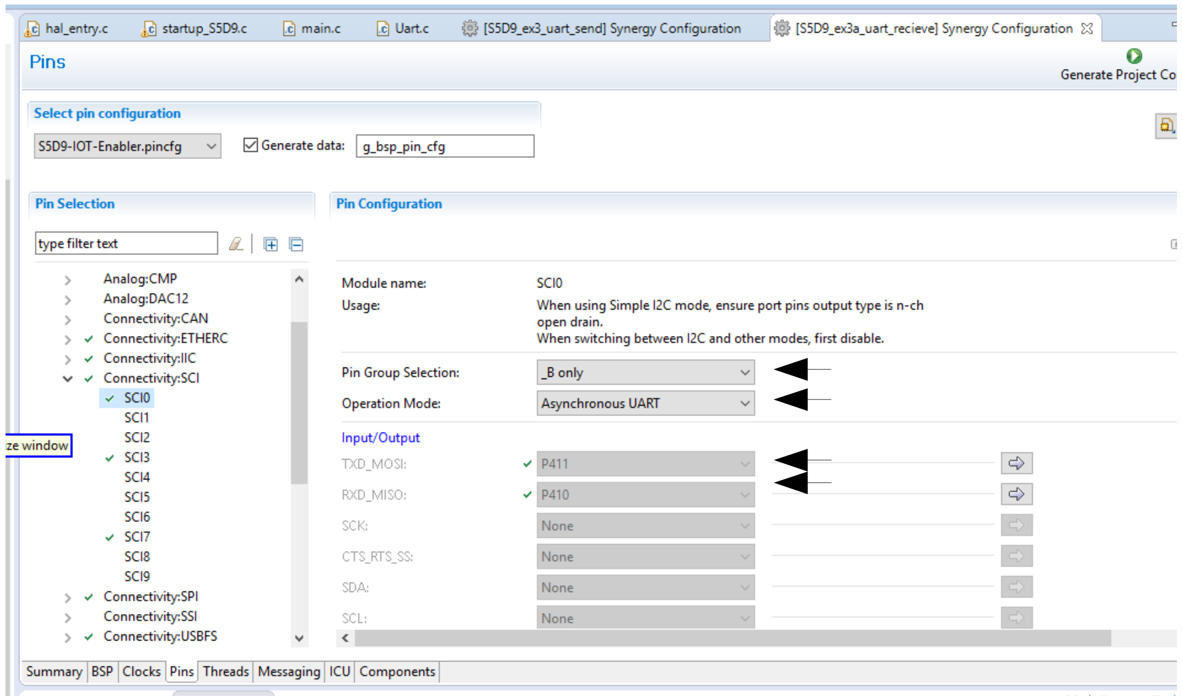
Grove A J3: UART to FTDI USB-to-UART Cable
Pin 1: Yellow RXD → Pin 4 Orange TXD
Pin 2: White TXD → Pin 5 Yellow RXD
Pin 3: Red 3V (No Connect)
Pin 4: Black GND → Pin 1 Black GND

Grove (UART)



Grove A
J3
P4_10/RXD0
P4_11/TXD0
+3V3
C8
0.1uF
GND
Header 4
UART3/GPIO
+3V3
RA14B
RA14A
4K70
P1_0/SCL1
P1_1/SDA1
C9
0.1uF
Grove B
J4
Header 4
I2C2/PWM
PMOD
J5
P2_5/SSLB0_A/CTS9_A
P2_3/MOSIB_A/TXD9_A
P2_2/MISOB_A/RXD9_A
P2_4/RSPCKB_A/SCK9_A
P7_8/IRQ11
P2_6/SSLB1_A
P2_7/SSLB2_A
P8_1
C10
0.1uF
+5V
+3V3
J6

Pin Configuration (SC0 Asy UART)



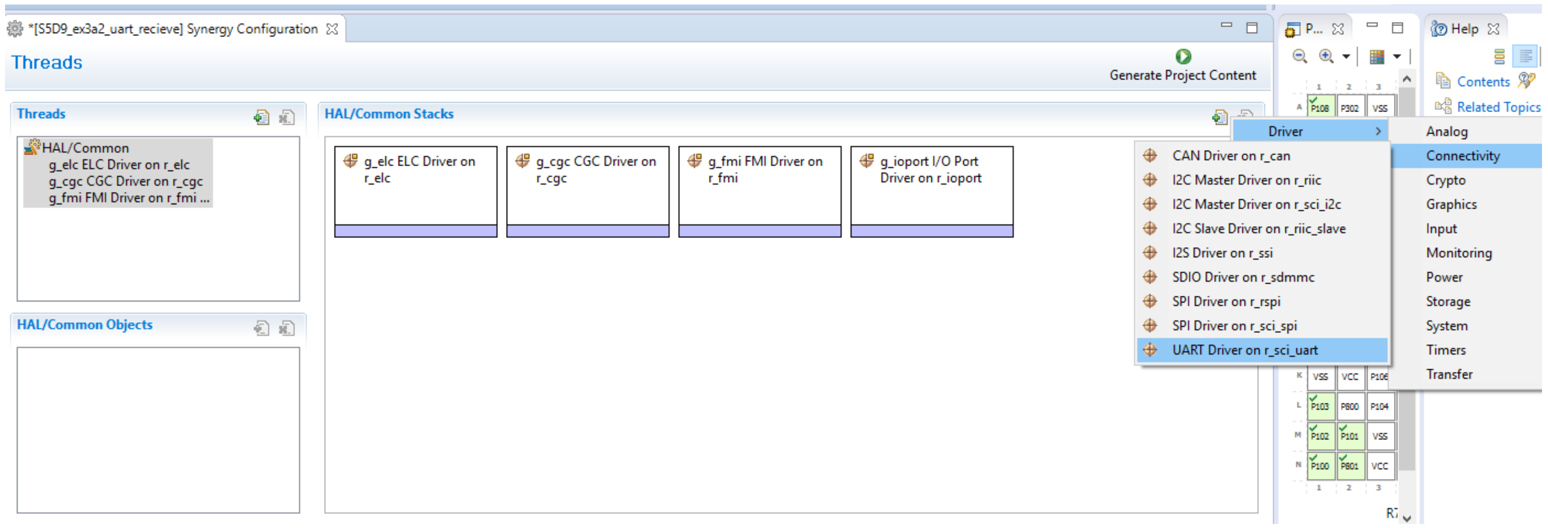
S5D9-IOT-Enabler.pincfg ☑ Generate data: g_bsp_pin_cfg

Module name:	SCI0
Usage:	When using Simple I2C mode, ensure port pins output type is n-ch open drain. When switching between I2C and other modes, first disable.
Pin Group Selection:	_B only
Operation Mode:	Asynchronous UART

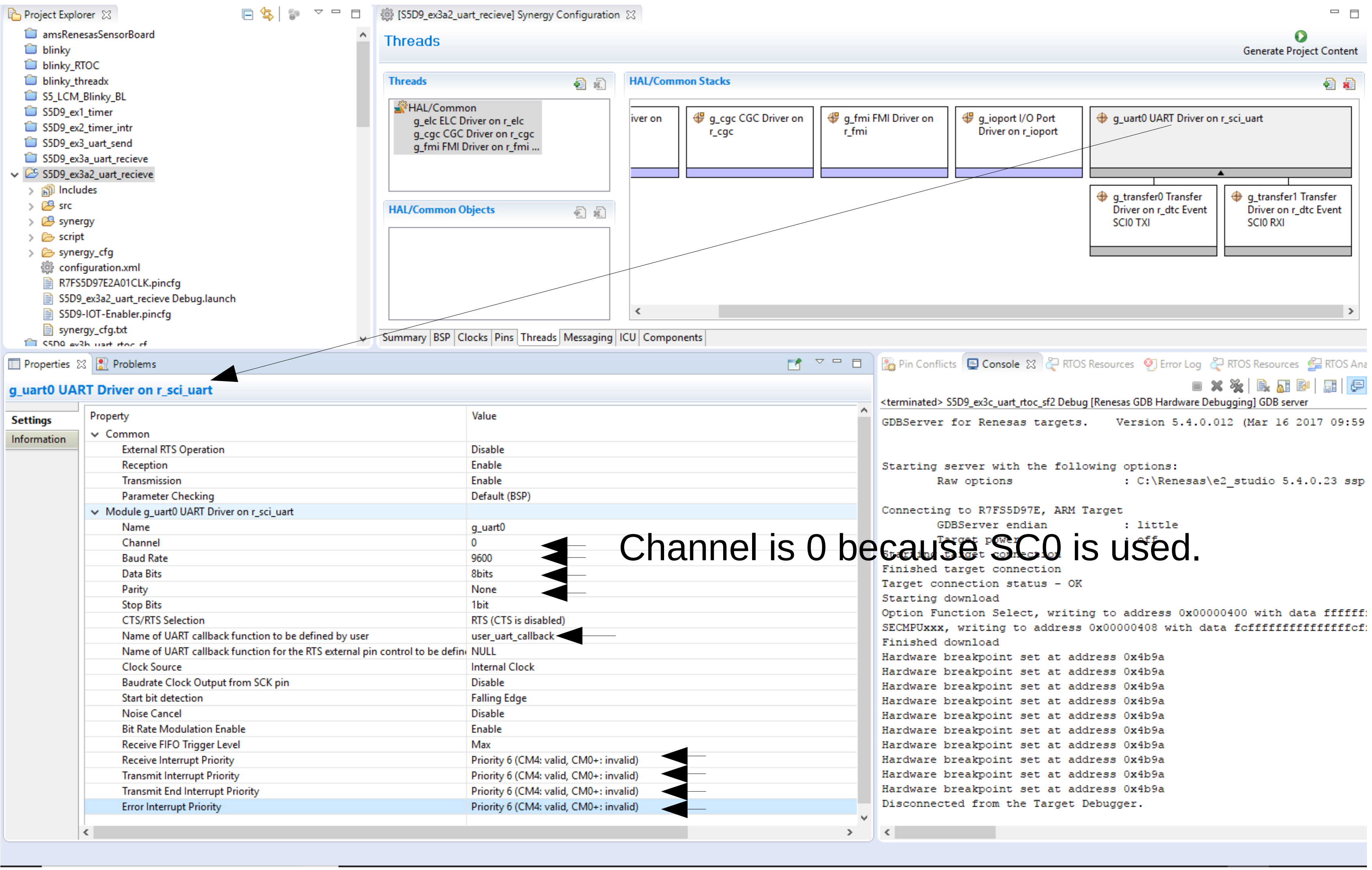
Input/Output

TXD_MOSI:	✓ P411
RXD_MISO:	✓ P410
SCK:	None
CTS_RTS_SS:	None
SDA:	None
SCL:	None

Select the UART Driver


*[S5D9_ex3a2_uart_recieve] Synergy Configuration
Threads
Generate Project Content
Threads
HAL/Common
g_elc ELC Driver on r_elc
g_cgc CGC Driver on r_cgc
g_fmi FMI Driver on r_fmi ...
HAL/Common Objects
HAL/Common Stacks
g_elc ELC Driver on r_elc
g_cgc CGC Driver on r_cgc
g_fmi FMI Driver on r_fmi
g_ioport I/O Port Driver on r_ioport
Driver
CAN Driver on r_can
I2C Master Driver on r_riic
I2C Master Driver on r_sci_i2c
I2C Slave Driver on r_riic_slave
I2S Driver on r_ssi
SDIO Driver on r_sdmmc
SPI Driver on r_rspi
SPI Driver on r_sci_spi
UART Driver on r_sci_uart
Analog
Connectivity
Crypto
Graphics
Input
Monitoring
Power
Storage
System
Timers
Transfer
Help
Contents
Related Topics

Configure the driver



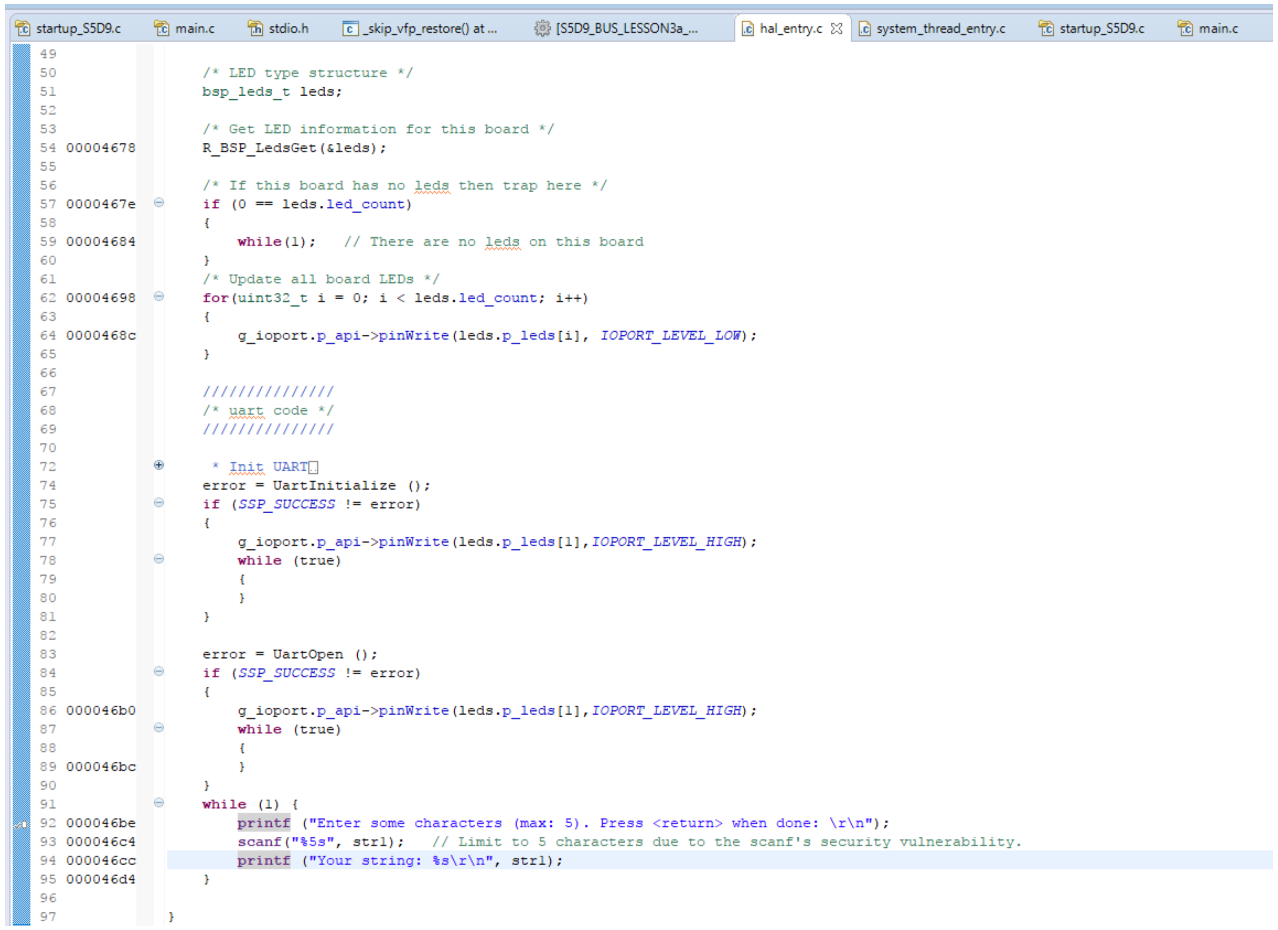
g_uart0 UART Driver on r_sci_uart

Property	Value
Common	
External RTS Operation	Disable
Reception	Enable
Transmission	Enable
Parameter Checking	Default (BSP)
Module g_uart0 UART Driver on r_sci_uart	
Name	g_uart0
Channel	0
Baud Rate	9600
Data Bits	8bits
Parity	None
Stop Bits	1bit
CTS/RTS Selection	RTS (CTS is disabled)
Name of UART callback function to be defined by user	user_uart_callback
Name of UART callback function for the RTS external pin control to be defin	NULL
Clock Source	Internal Clock
Baudrate Clock Output from SCK pin	Disable
Start bit detection	Falling Edge
Noise Cancel	Disable
Bit Rate Modulation Enable	Enable
Receive FIFO Trigger Level	Max
Receive Interrupt Priority	Priority 6 (CM4: valid, CM0+: invalid)
Transmit Interrupt Priority	Priority 6 (CM4: valid, CM0+: invalid)
Transmit End Interrupt Priority	Priority 6 (CM4: valid, CM0+: invalid)
Error Interrupt Priority	Priority 6 (CM4: valid, CM0+: invalid)

```
<terminated> S5D9_ex3c_uart_rtoc_sf2 Debug [Renesas GDB Hardware Debugging] GDB server
GDBServer for Renesas targets.    Version 5.4.0.012 (Mar 16 2017 09:59


Starting server with the following options:
        Raw options                  : C:\Renesas\e2_studio 5.4.0.23 ssp

Connecting to R7FS5D97E, ARM Target
        GDBServer endian             : little
        Target power                 : off
Starting target connection
Finished target connection
Target connection status - OK
Starting download
Option Function Select, writing to address 0x00000400 with data ffffff
SECMPUxxx, writing to address 0x00000408 with data fcfffffffffffffcf
Finished download
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Hardware breakpoint set at address 0x4b9a
Disconnected from the Target Debugger.
```

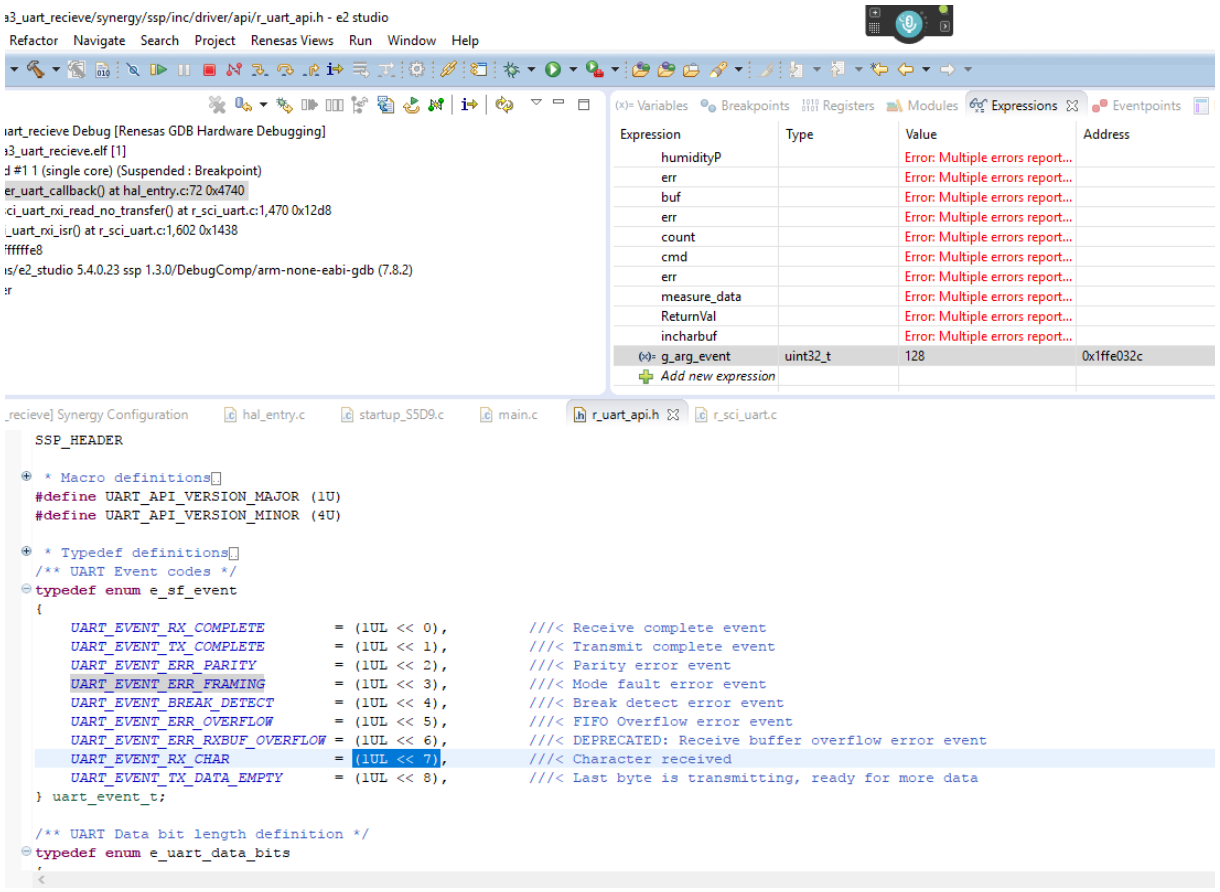
Channel is 0 because SC0 is used.

Main Loop



```
                 /* LED type structure */
                 bsp_leds_t leds;

                 /* Get LED information for this board */
00004678         R_BSP_LedsGet(&leds);

                 /* If this board has no leds then trap here */
0000467e         if (0 == leds.led_count)
                 {
00004684             while(1);    // There are no leds on this board
                 }
                 /* Update all board LEDs */
00004698         for(uint32_t i = 0; i < leds.led_count; i++)
                 {
0000468c             g_ioport.p_api->pinWrite(leds.p_leds[i], IOPORT_LEVEL_LOW);
                 }

                 ///////////////
                 /* uart code */
                 ///////////////

                  * Init UART
                 error = UartInitialize ();
                 if (SSP_SUCCESS != error)
                 {
                     g_ioport.p_api->pinWrite(leds.p_leds[1],IOPORT_LEVEL_HIGH);
                     while (true)
                     {
                     }
                 }

                 error = UartOpen ();
                 if (SSP_SUCCESS != error)
                 {
000046b0             g_ioport.p_api->pinWrite(leds.p_leds[1],IOPORT_LEVEL_HIGH);
                     while (true)
                     {
000046bc             }
                 }
                 while (1) {
000046be             printf ("Enter some characters (max: 5). Press <return> when done: \r\n");
000046c4             scanf("%5s", str1);   // Limit to 5 characters due to the scanf's security vulnerability.
000046cc             printf ("Your string: %s\r\n", str1);
000046d4         }

             }
```

After each char is received, an event flag is set to 128 (UART_EVENT_RX_CHAR)

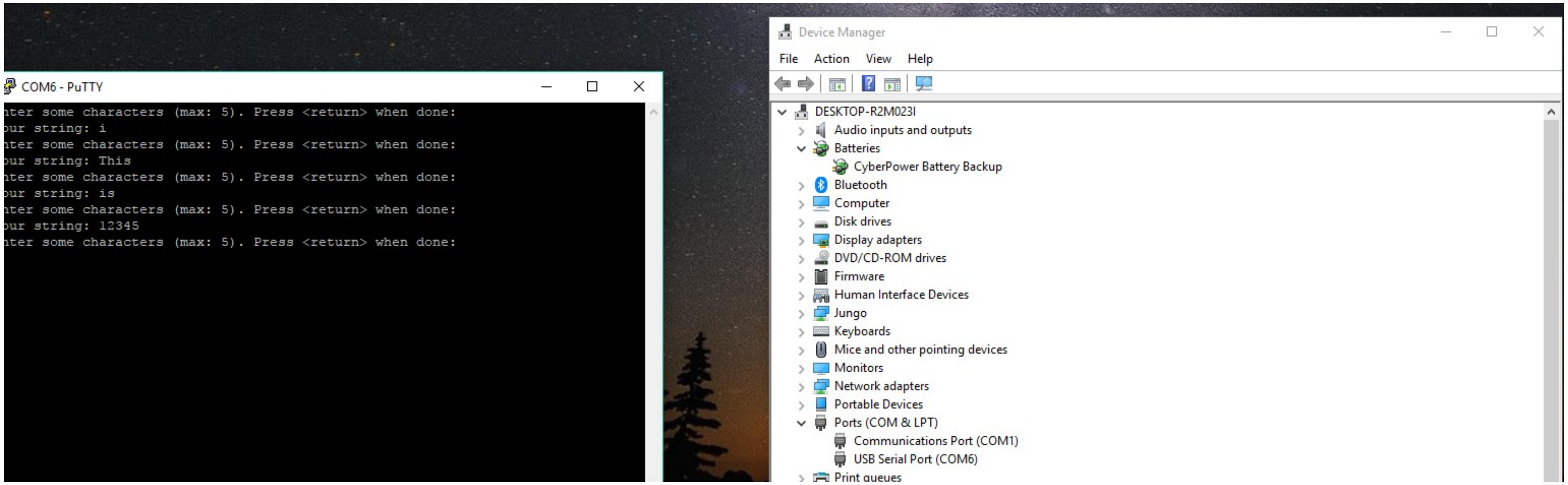

a3_uart_recieve/synergy/ssp/inc/driver/api/r_uart_api.h - e2 studio
Refactor Navigate Search Project Renesas Views Run Window Help
(x)= Variables Breakpoints Registers Modules Expressions Eventpoints
recieve] Synergy Configuration hal_entry.c startup_S5D9.c main.c r_uart_api.h r_sci_uart.c

```
uart_recieve Debug [Renesas GDB Hardware Debugging]
a3_uart_recieve.elf [1]
d #1 1 (single core) (Suspended : Breakpoint)
er_uart_callback() at hal_entry.c:72 0x4740
sci_uart_rxi_read_no_transfer() at r_sci_uart.c:1,470 0x12d8
i_uart_rxi_isr() at r_sci_uart.c:1,602 0x1438
ffffffe8
as/e2_studio 5.4.0.23 ssp 1.3.0/DebugComp/arm-none-eabi-gdb (7.8.2)
er
```

Expression	Type	Value	Address
humidityP		Error: Multiple errors report...	
err		Error: Multiple errors report...	
buf		Error: Multiple errors report...	
err		Error: Multiple errors report...	
count		Error: Multiple errors report...	
cmd		Error: Multiple errors report...	
err		Error: Multiple errors report...	
measure_data		Error: Multiple errors report...	
ReturnVal		Error: Multiple errors report...	
incharbuf		Error: Multiple errors report...	
(x)= g_arg_event	uint32_t	128	0x1ffe032c
Add new expression			

```
SSP_HEADER

 * Macro definitions
#define UART_API_VERSION_MAJOR (1U)
#define UART_API_VERSION_MINOR (4U)

 * Typedef definitions
/** UART Event codes */
typedef enum e_sf_event
{
    UART_EVENT_RX_COMPLETE        = (1UL << 0),       ///< Receive complete event
    UART_EVENT_TX_COMPLETE        = (1UL << 1),       ///< Transmit complete event
    UART_EVENT_ERR_PARITY         = (1UL << 2),       ///< Parity error event
    UART_EVENT_ERR_FRAMING        = (1UL << 3),       ///< Mode fault error event
    UART_EVENT_BREAK_DETECT       = (1UL << 4),       ///< Break detect error event
    UART_EVENT_ERR_OVERFLOW       = (1UL << 5),       ///< FIFO Overflow error event
    UART_EVENT_ERR_RXBUF_OVERFLOW = (1UL << 6),       ///< DEPRECATED: Receive buffer overflow error event
    UART_EVENT_RX_CHAR            = (1UL << 7),       ///< Character received
    UART_EVENT_TX_DATA_EMPTY      = (1UL << 8),       ///< Last byte is transmitting, ready for more data
} uart_event_t;

/** UART Data bit length definition */
typedef enum e_uart_data_bits
```

Use Device Manager to check for the COM port #.



```
nter some characters (max: 5). Press <return> when done:
our string: i
nter some characters (max: 5). Press <return> when done:
our string: This
nter some characters (max: 5). Press <return> when done:
our string: is
nter some characters (max: 5). Press <return> when done:
our string: 12345
nter some characters (max: 5). Press <return> when done:
```