S5D9 SPI Bus Example (SCI_SPI Framework Version)

By
Michael Li

(2/2/2018)

https://www.miketechuniverse.com

E2 Studio 5.4.0.023
SSP 1.3.0
ThreadX ROTC

s5d9_lab_spi (PMOD)

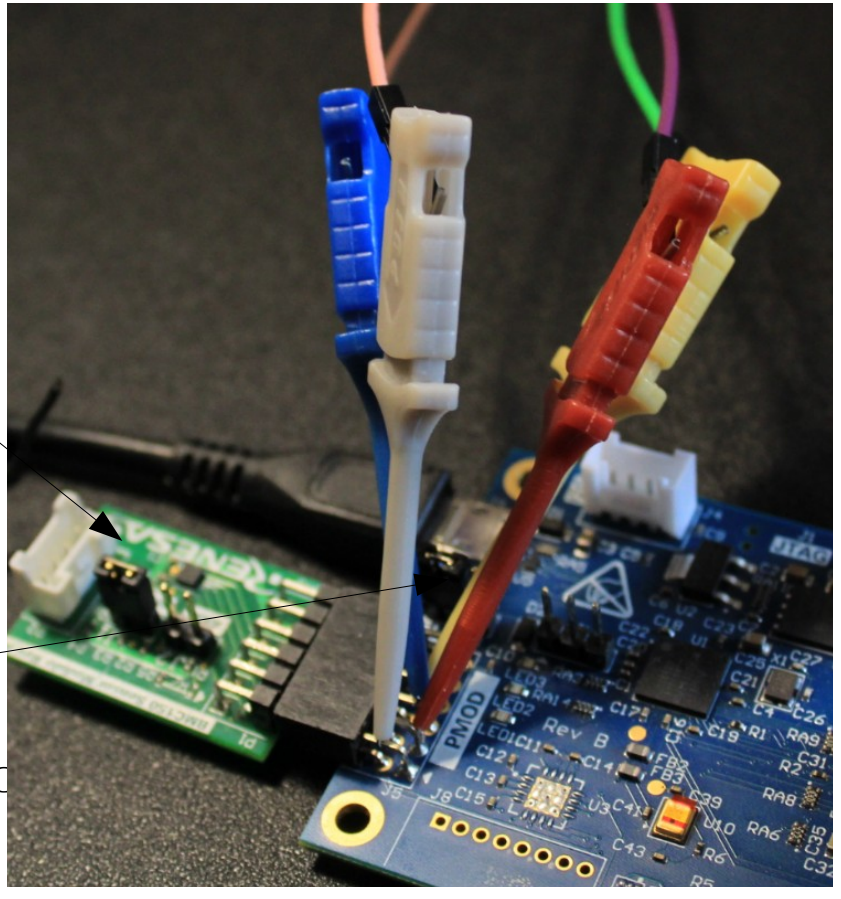
1. Use PMOD J5
2. SSP 1.3.0
3. E2 studio Version: 5.4.0.023

Project Summary

Board:	S5D9_IOT_ENABLER
Device:	R7FS5D97E2A01CLK
Toolchain:	GCC ARM Embedded
Toolchain Version:	4.9.3.20150529
SSP Version:	1.3.0

Selected software components:

Jumper close for SPI Mode

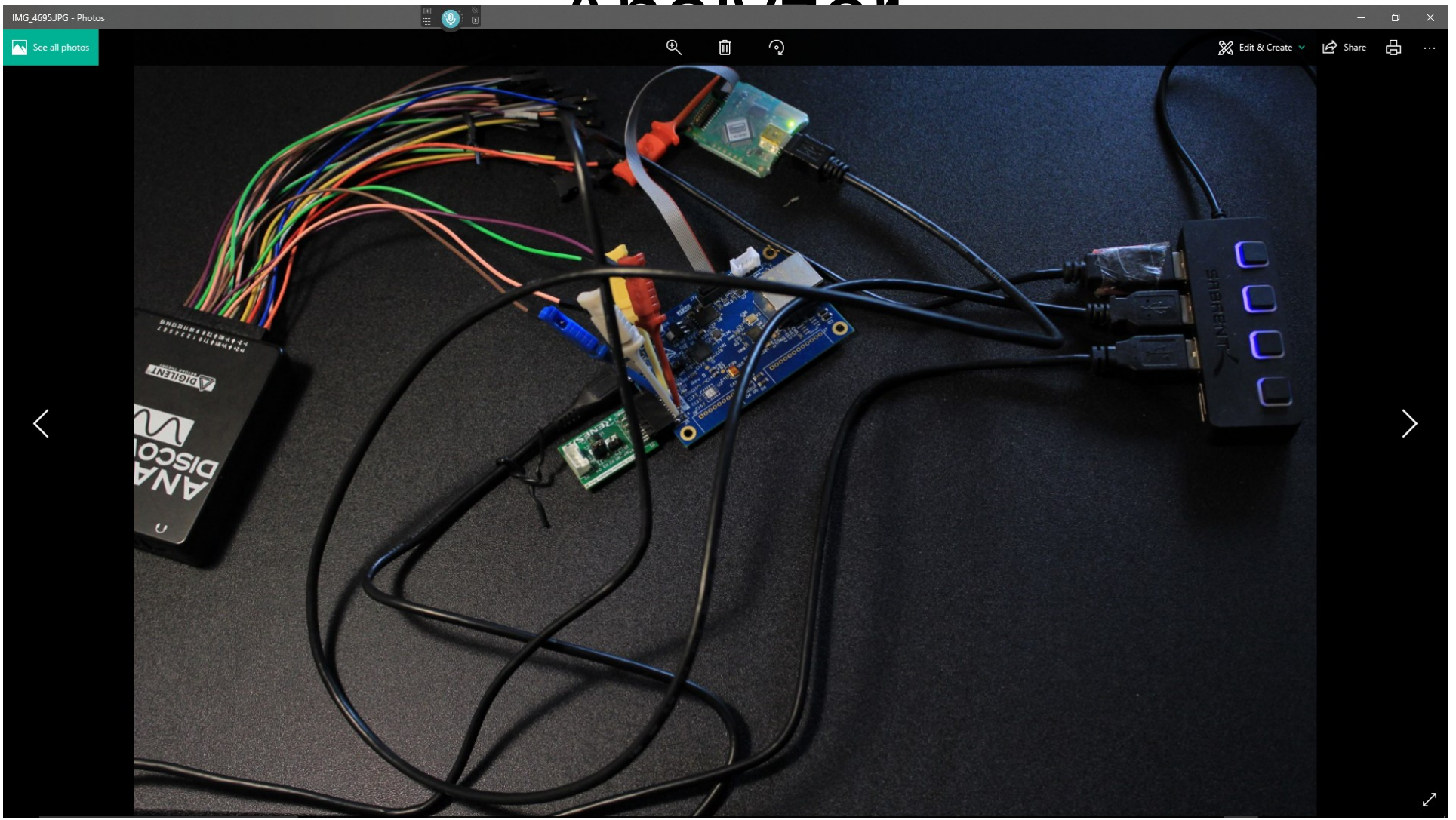


D0 P1 CSB
D2 P2 MOSI
D3 P3 MISO
D1 P4 CK
P5 GND
P6 VCC = 3.3V (due to JMP)

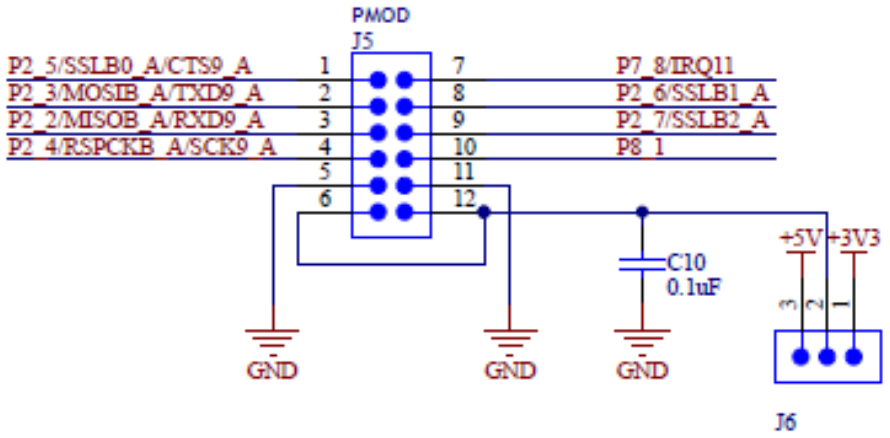
SPI Example by Michael C

Px = PMOD pin, Dx = Scope pin

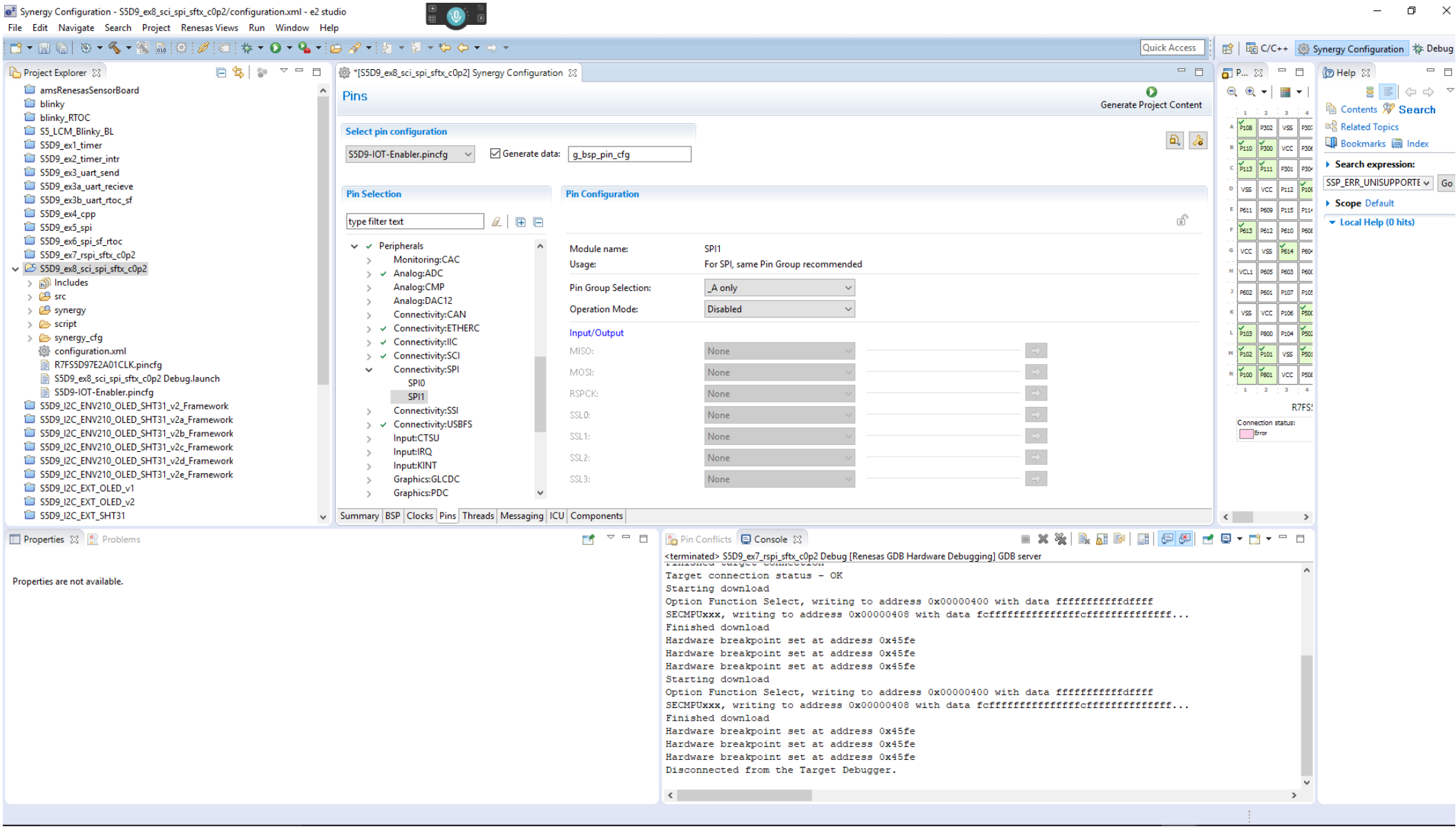
The Hardware Setup with The Logic Analyzer



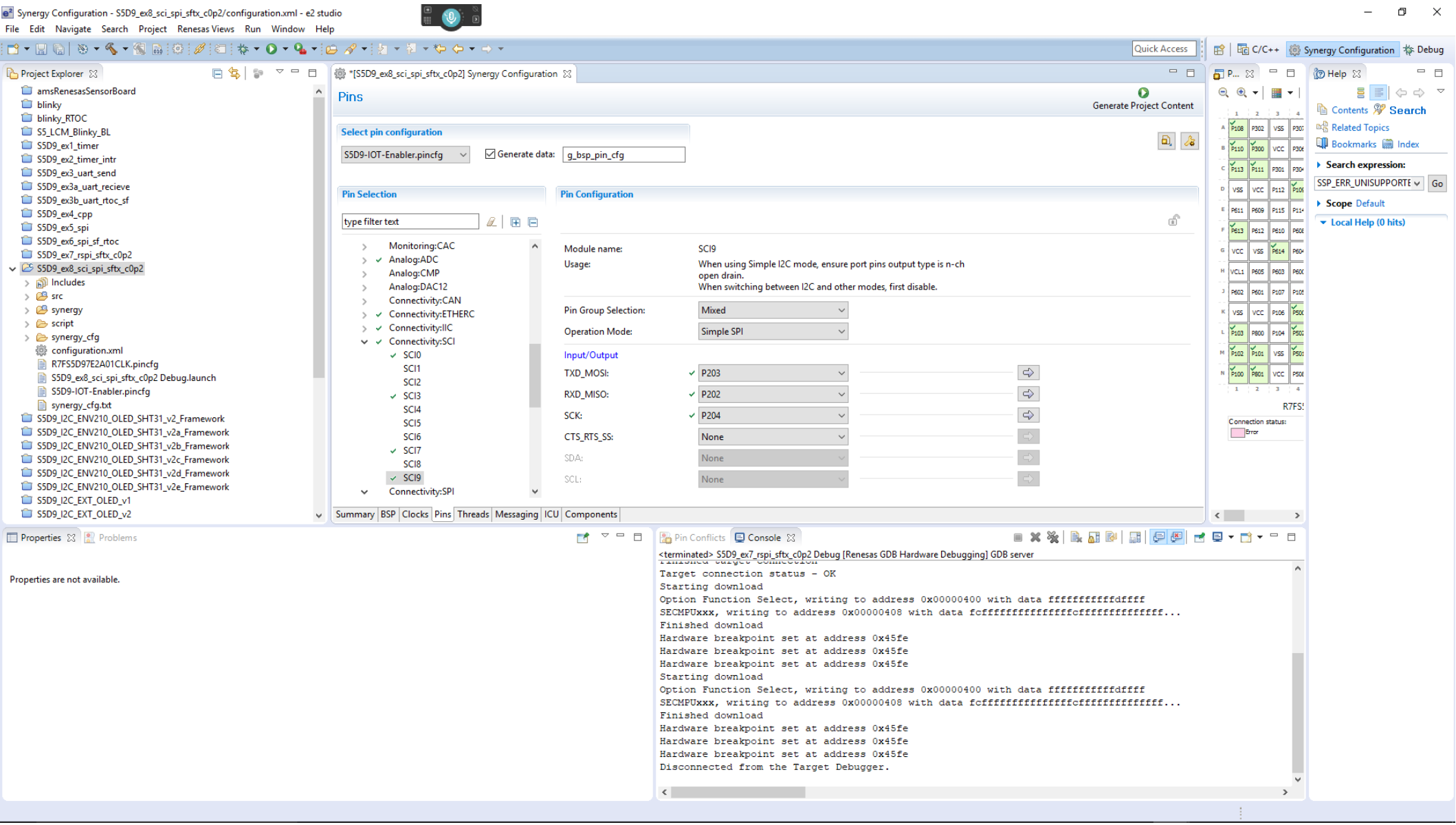
PMOD Schematic



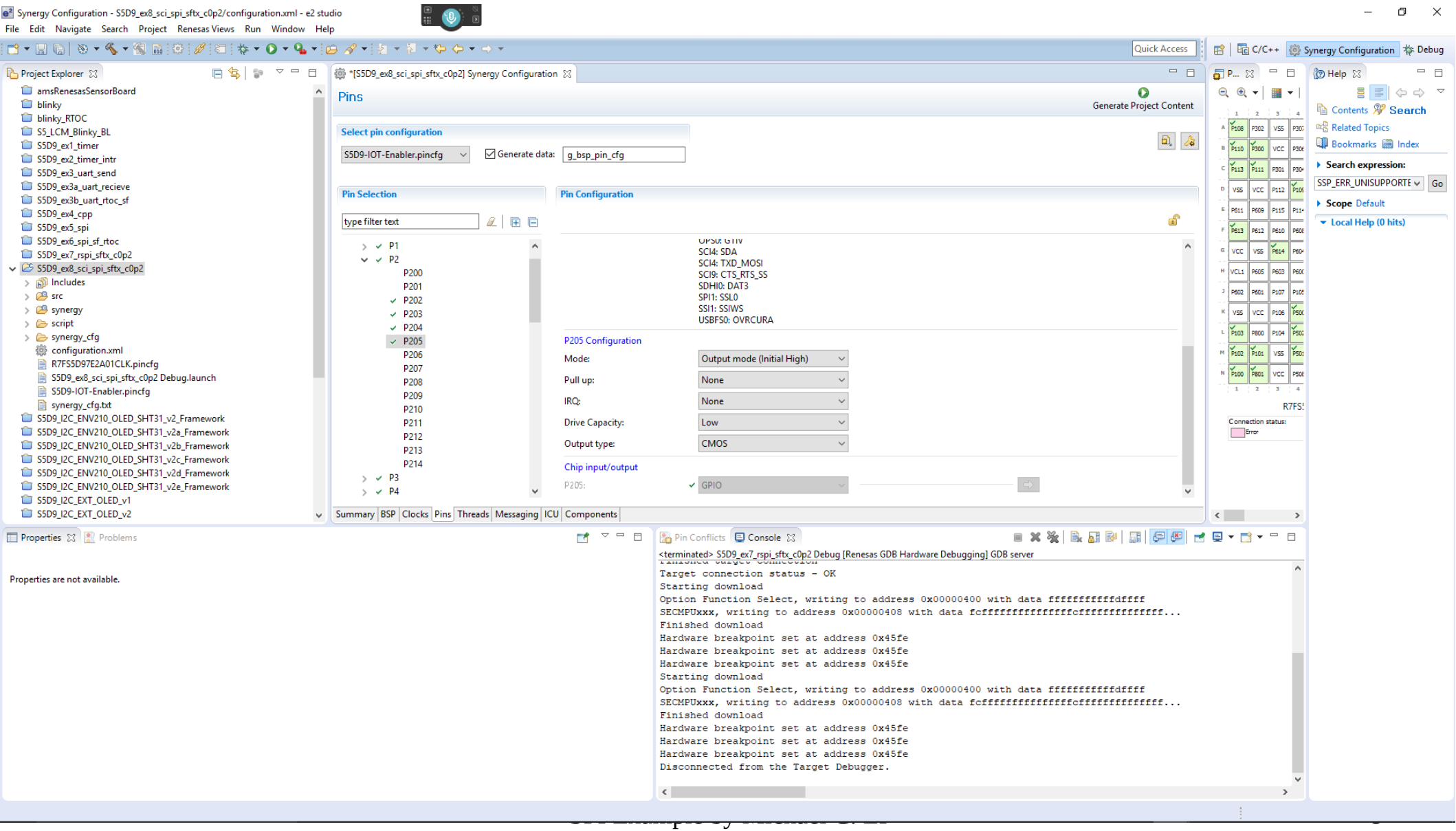
Disable SPI1



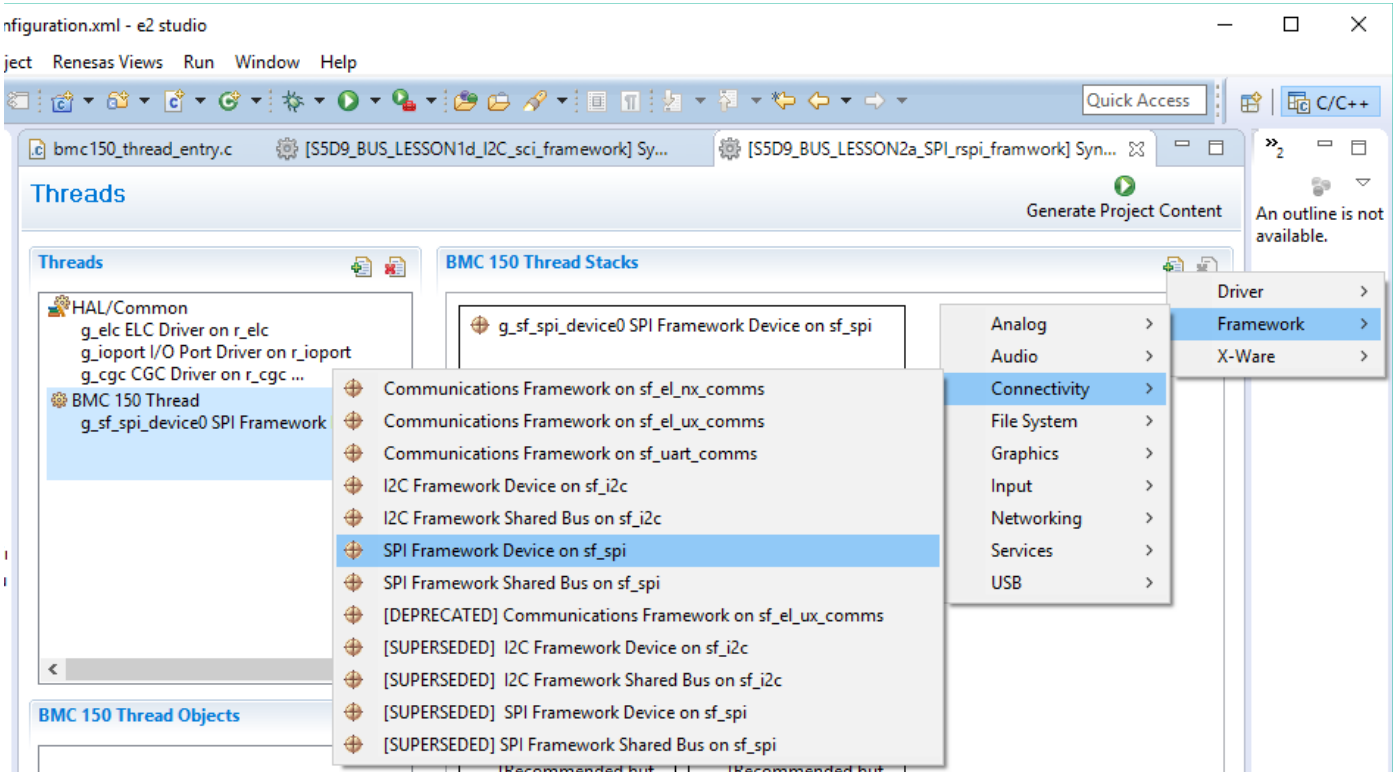
Eanble SCI9


Synergy Configuration - S5D9_ex8_sci_spi_sftx_c0p2/configuration.xml - e2 studio
File Edit Navigate Search Project Renesas Views Run Window Help
Quick Access
C/C++ Synergy Configuration Debug
Project Explorer
amsRenesasSensorBoard
blinky
blinky_RTOC
S5_LCM_Blinky_BL
S5D9_ex1_timer
S5D9_ex2_timer_intr
S5D9_ex3_uart_send
S5D9_ex3a_uart_recieve
S5D9_ex3b_uart_rtoc_sf
S5D9_ex4_cpp
S5D9_ex5_spi
S5D9_ex6_spi_sf_rtoc
S5D9_ex7_rspi_sftx_c0p2
S5D9_ex8_sci_spi_sftx_c0p2
Includes
src
synergy
script
synergy_cfg
configuration.xml
R7FS5D97E2A01CLK.pincfg
S5D9_ex8_sci_spi_sftx_c0p2 Debug.launch
S5D9-IOT-Enabler.pincfg
synergy_cfg.txt
S5D9_I2C_ENV210_OLED_SHT31_v2_Framework
S5D9_I2C_ENV210_OLED_SHT31_v2a_Framework
S5D9_I2C_ENV210_OLED_SHT31_v2b_Framework
S5D9_I2C_ENV210_OLED_SHT31_v2c_Framework
S5D9_I2C_ENV210_OLED_SHT31_v2d_Framework
S5D9_I2C_ENV210_OLED_SHT31_v2e_Framework
S5D9_I2C_EXT_OLED_v1
S5D9_I2C_EXT_OLED_v2
*[S5D9_ex8_sci_spi_sftx_c0p2] Synergy Configuration
Pins
Generate Project Content
Select pin configuration
S5D9-IOT-Enabler.pincfg
Generate data:
g_bsp_pin_cfg
Pin Selection
type filter text
Monitoring:CAC
Analog:ADC
Analog:CMP
Analog:DAC12
Connectivity:CAN
Connectivity:ETHERC
Connectivity:IIC
Connectivity:SCI
SCI0
SCI1
SCI2
SCI3
SCI4
SCI5
SCI6
SCI7
SCI8
SCI9
Connectivity:SPI
Pin Configuration
Module name: SCI9
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: Mixed
Operation Mode: Simple SPI
Input/Output
TXD_MOSI: P203
RXD_MISO: P202
SCK: P204
CTS_RTS_SS: None
SDA: None
SCL: None
Summary BSP Clocks Pins Threads Messaging ICU Components
Help
Contents Search
Related Topics
Bookmarks Index
Search expression:
SSP_ERR_UNISUPPORTE
Go
Scope Default
Local Help (0 hits)
Properties
Problems
Properties are not available.
Pin Conflicts
Console
<terminated> S5D9_ex7_rspi_sftx_c0p2 Debug [Renesas GDB Hardware Debugging] GDB server
Target connection status - OK
Starting download
Option Function Select, writing to address 0x00000400 with data ffffffffffdffff
SECMPUxxx, writing to address 0x00000408 with data fcffffffffffffffcffffffffffffff...
Finished download
Hardware breakpoint set at address 0x45fe
Hardware breakpoint set at address 0x45fe
Hardware breakpoint set at address 0x45fe
Starting download
Option Function Select, writing to address 0x00000400 with data ffffffffffdffff
SECMPUxxx, writing to address 0x00000408 with data fcffffffffffffffcffffffffffffff...
Finished download
Hardware breakpoint set at address 0x45fe
Hardware breakpoint set at address 0x45fe
Hardware breakpoint set at address 0x45fe
Disconnected from the Target Debugger.

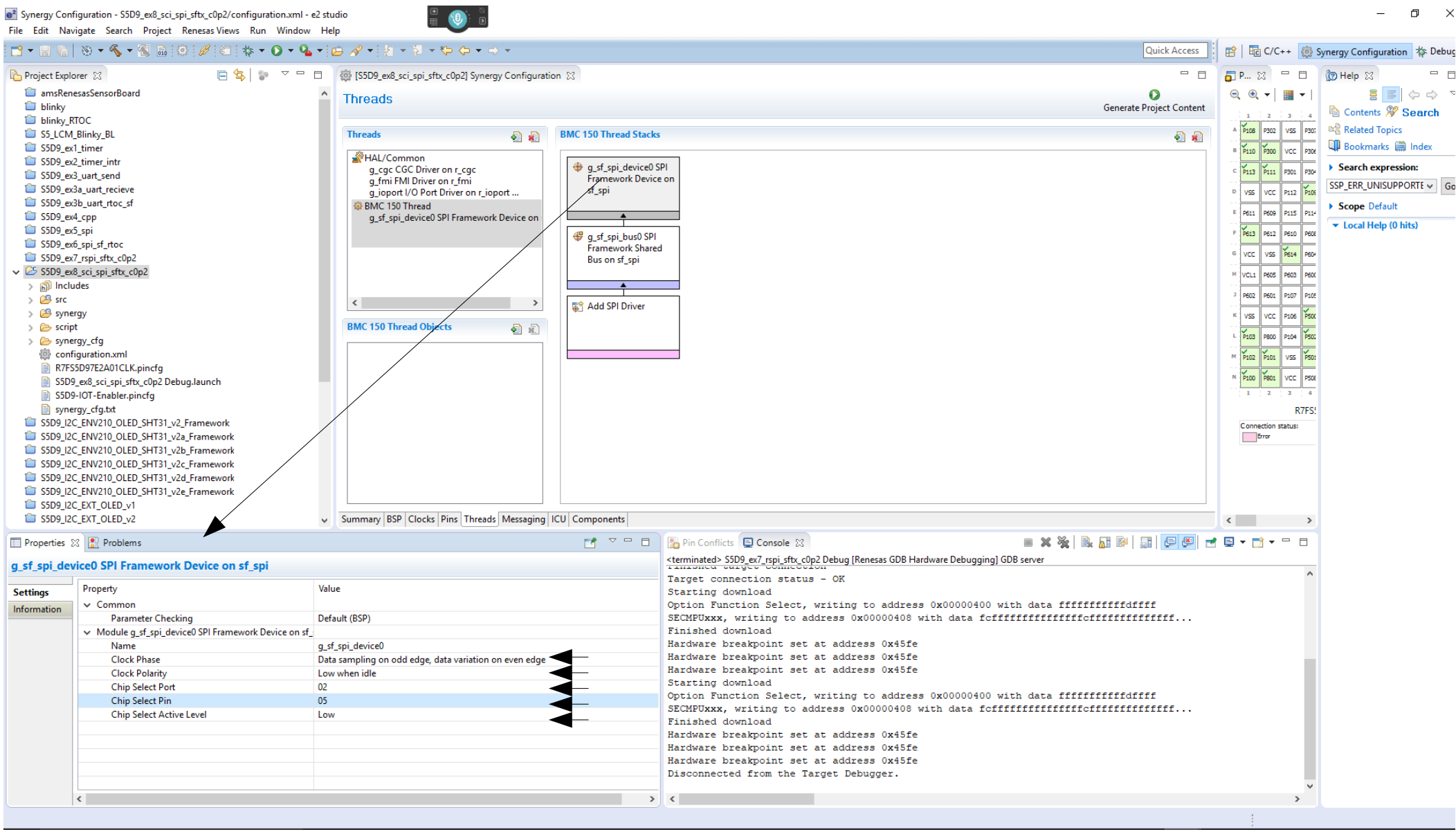
SSL0 P205 (GPIO)



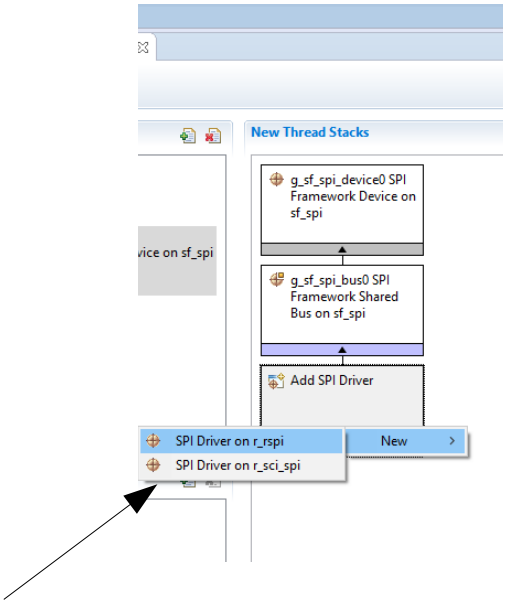
Create a new Thread and SPI Framework Device



Configure Chip Select

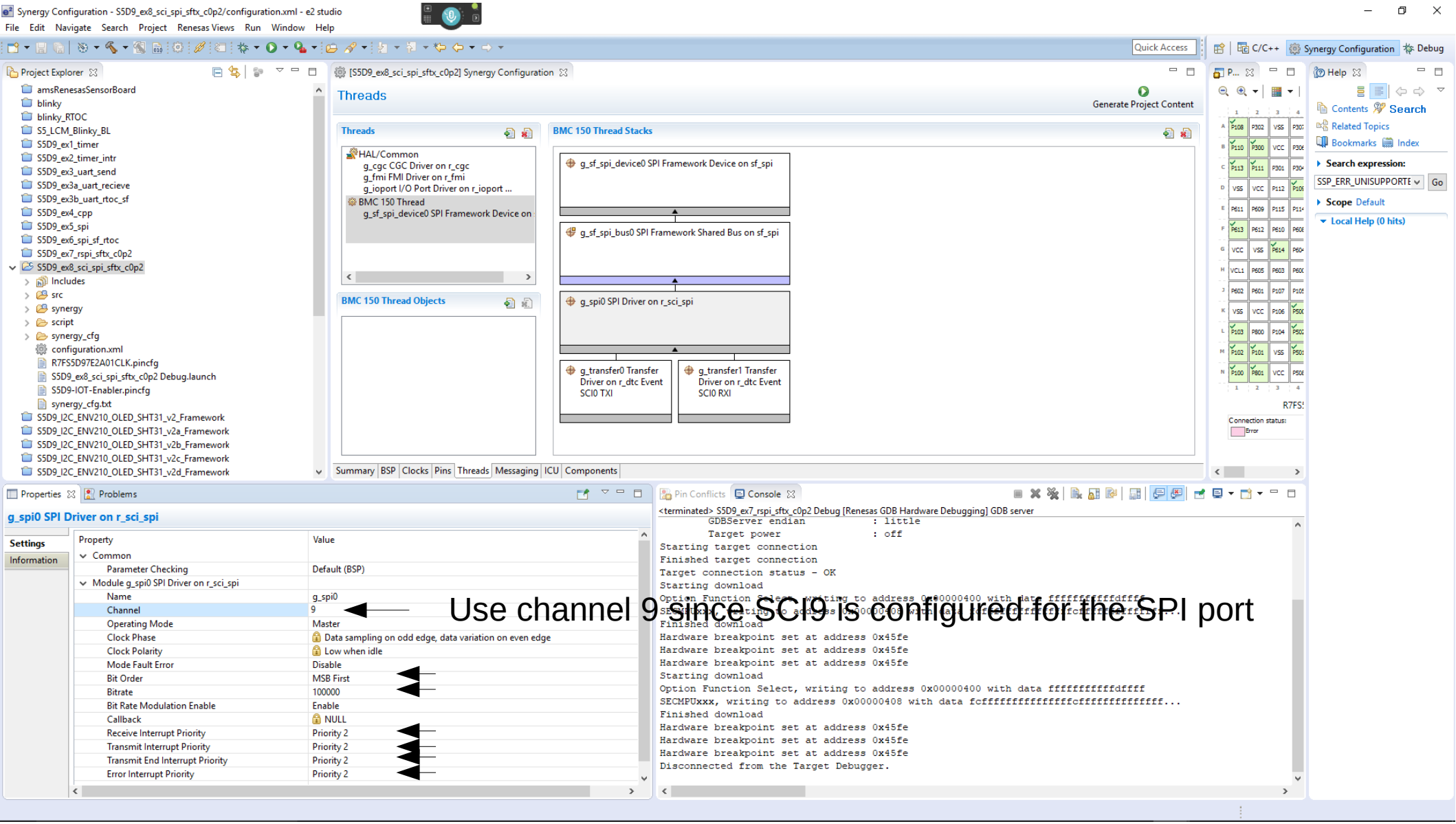


Choose r_sci_spi driver for new



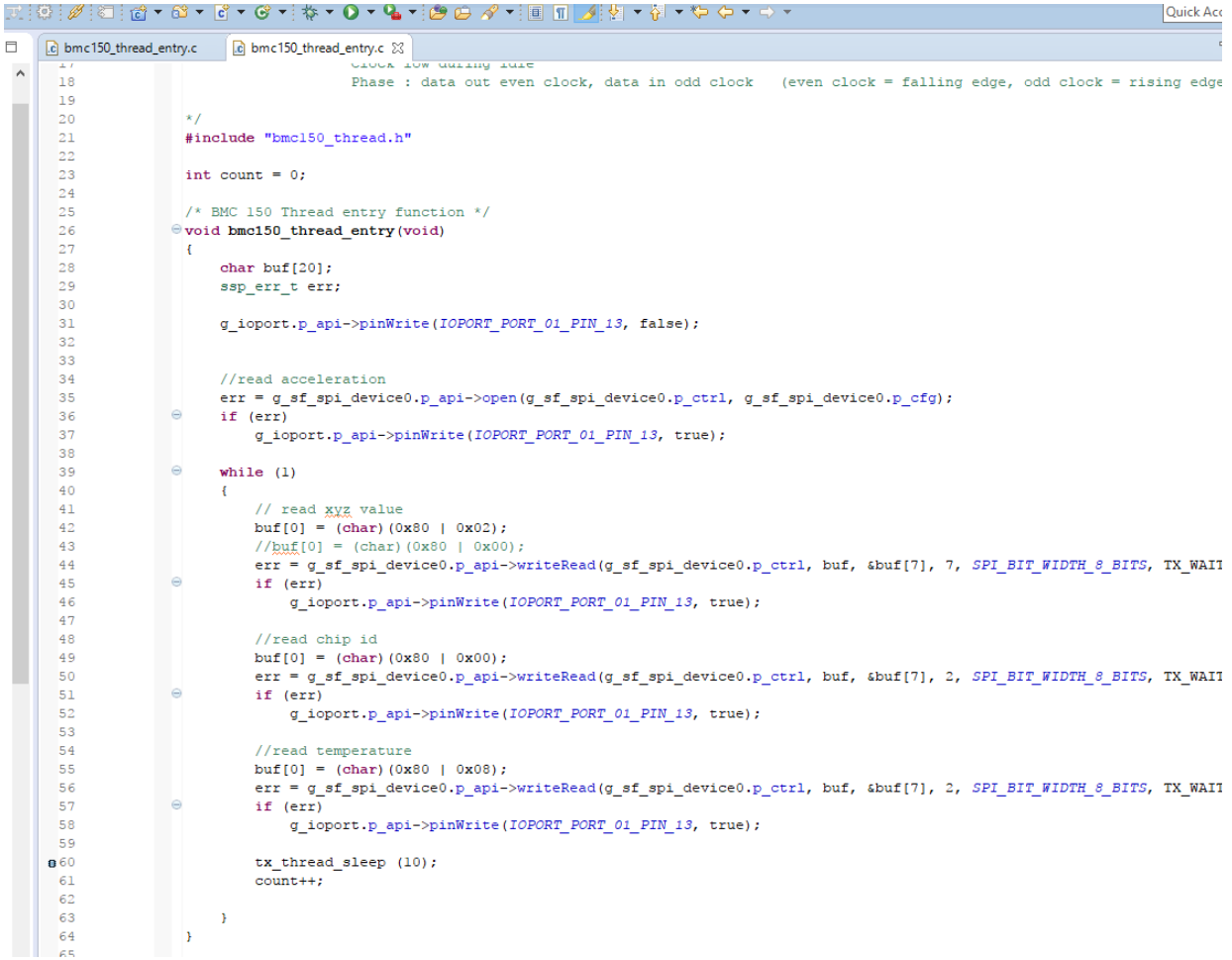
Choose the bottom one

Configure the r_sci_spi driver



Use channel 9 since SCI9 is configured for the SPI port

Main Code


bmc150_thread_entry.c
bmc150_thread_entry.c
Phase : data out even clock, data in odd clock (even clock = falling edge, odd clock = rising edge
*/
#include "bmc150_thread.h"
int count = 0;
/* BMC 150 Thread entry function */
void bmc150_thread_entry(void)
{
char buf[20];
ssp_err_t err;
g_ioport.p_api->pinWrite(IOPORT_PORT_01_PIN_13, false);
//read acceleration
err = g_sf_spi_device0.p_api->open(g_sf_spi_device0.p_ctrl, g_sf_spi_device0.p_cfg);
if (err)
g_ioport.p_api->pinWrite(IOPORT_PORT_01_PIN_13, true);
while (1)
{
// read xyz value
buf[0] = (char)(0x80 | 0x02);
//buf[0] = (char)(0x80 | 0x00);
err = g_sf_spi_device0.p_api->writeRead(g_sf_spi_device0.p_ctrl, buf, &buf[7], 7, SPI_BIT_WIDTH_8_BITS, TX_WAIT
if (err)
g_ioport.p_api->pinWrite(IOPORT_PORT_01_PIN_13, true);
//read chip id
buf[0] = (char)(0x80 | 0x00);
err = g_sf_spi_device0.p_api->writeRead(g_sf_spi_device0.p_ctrl, buf, &buf[7], 2, SPI_BIT_WIDTH_8_BITS, TX_WAIT
if (err)
g_ioport.p_api->pinWrite(IOPORT_PORT_01_PIN_13, true);
//read temperature
buf[0] = (char)(0x80 | 0x08);
err = g_sf_spi_device0.p_api->writeRead(g_sf_spi_device0.p_ctrl, buf, &buf[7], 2, SPI_BIT_WIDTH_8_BITS, TX_WAIT
if (err)
g_ioport.p_api->pinWrite(IOPORT_PORT_01_PIN_13, true);
tx_thread_sleep (10);
count++;
}
}

Read ID Example (0xFA)