

Avnet Visible Things I2C Example

by

Michael C. Li

<https://www.miketechuniverse.com/>

4/5/2018

e2 Studio Version: 5.3.1.002

Project Summary

Board:	gw002_rev1_3
Device:	R7FS7G27H2A01CBD
Toolchain:	GCC ARM Embedded
Toolchain Version:	4.9.3.20150529
SSP Version:	1.2.0

PMOD Configuration for this example

Avnet Visible Things Platform

User's Manual

Gateway PMOD	I2C	Type 1 GPIO	Type 2 SPI	Type 2A Expanded SPI	Type 3 UART	Type 4 UART	Type 4A Expanded UART	Type 5 H- Bridge	Type 6 Dual H- Bridge
1	Y	Y	Y	Y	Y	Y	Y	N	N
2	Y	Y	Y	Y	Y	Y	Y	N	N
3	Y	Y	Y	Y	Y	Y	Y	N	N
4	Y	Y	Y	Y	Y	Y	Y	N	N
5	N	Y	Y	Y	N	Y	Y	Y	Y
6	Y	Y	N	N	N	N	N	N	N

Table 3 – Pmod Compatibility Chart

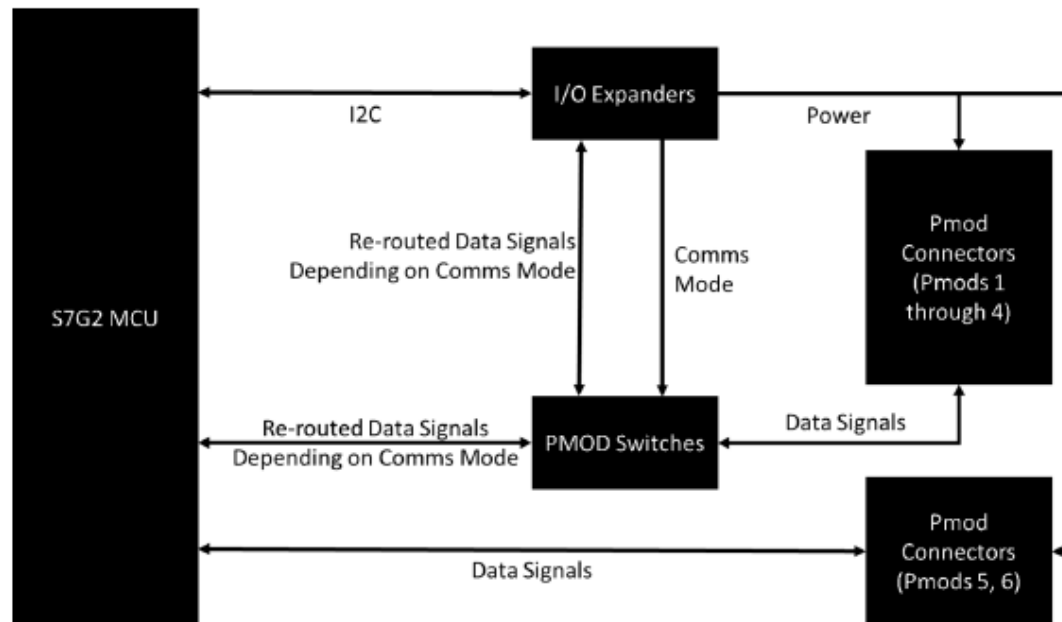


Figure 30 – Pmod, I/O Expander Interconnections Diagram

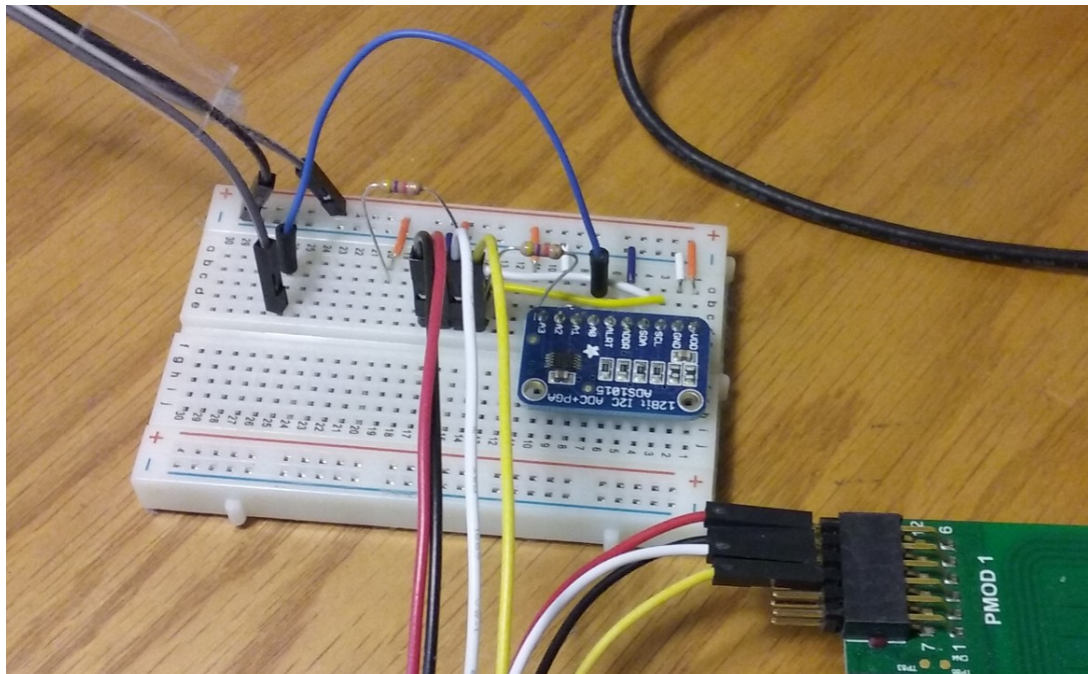
Hardware Setup



Add two 4.7k pullup on the daughter board.

PMOD1	ADS1015
Pin 1 NC	
Pin 2 NC	
Pin 3 SCL	SCL+4.7k pullup (Yellow)
Pin 4 SDA	SDA+4.7k Pullup (White)
Pin 5 GND	GND (Black)
Pin 6 VCC	VCC (Orange)

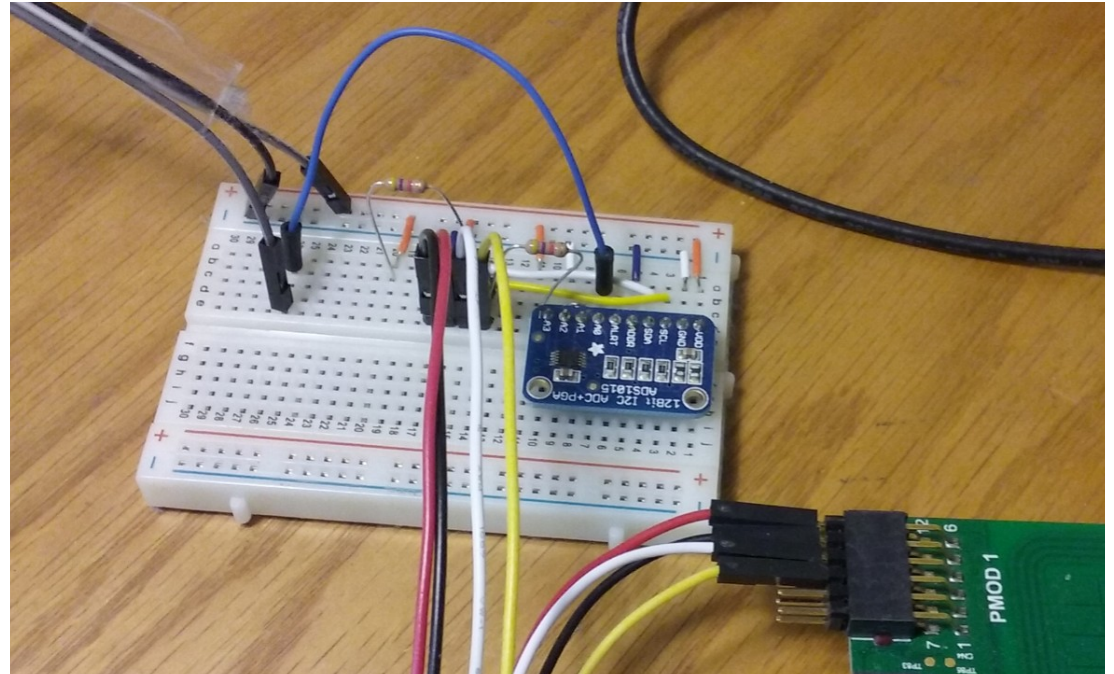
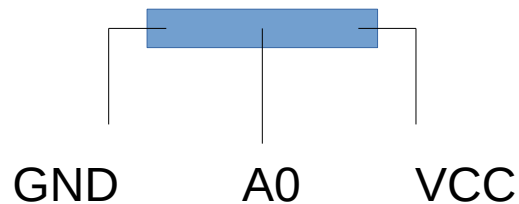
* NC = No Connect



Connection Setup for ADC Module

ADS1015
ADDR = GND
Alert = float (not used)
A0 = Potentiometer
A1/2/3 = GND (not used)

Potentiometer



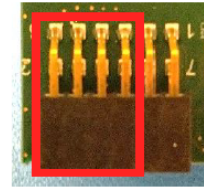
PMOD1 Pins

I2C Comms Mode

The I2C Comms Mode can be used for both I2C standard and UART Type 3 standard Pmods.

I2C Pmods

I2C Pmods only have 8 pins instead of the 12 provided by the Gateway's connector. Insert these Pmods into the left-most side of the connector.

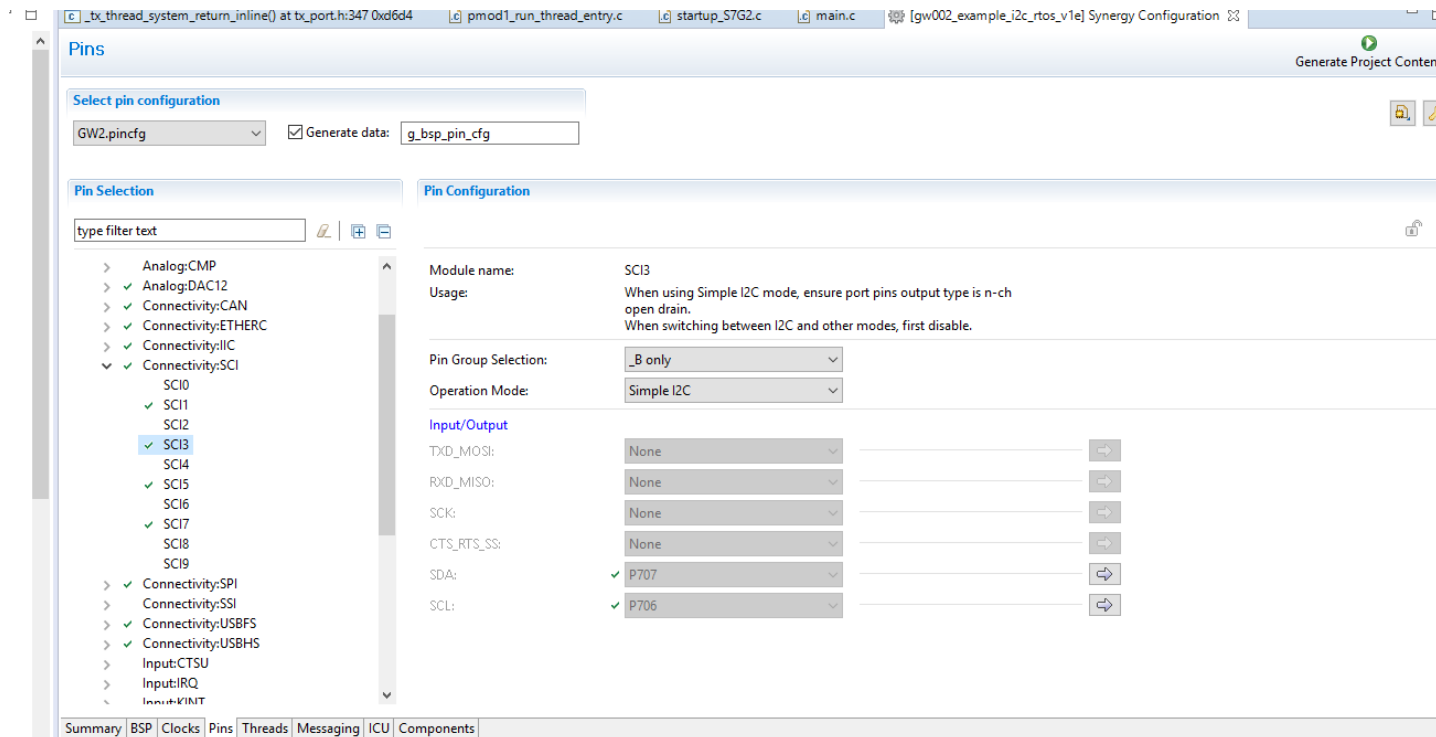


Pmod		S7G2	
Pin	Pin Function	Port Pin	Pin Function
1	SCL	PA02	SDA
2	SCL	PA02	SDA
3	SDA	PA03	SCL
4	SDA	PA03	SCL
5	GND	GND	GND
6	GND	GND	GND
7	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)
8	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)

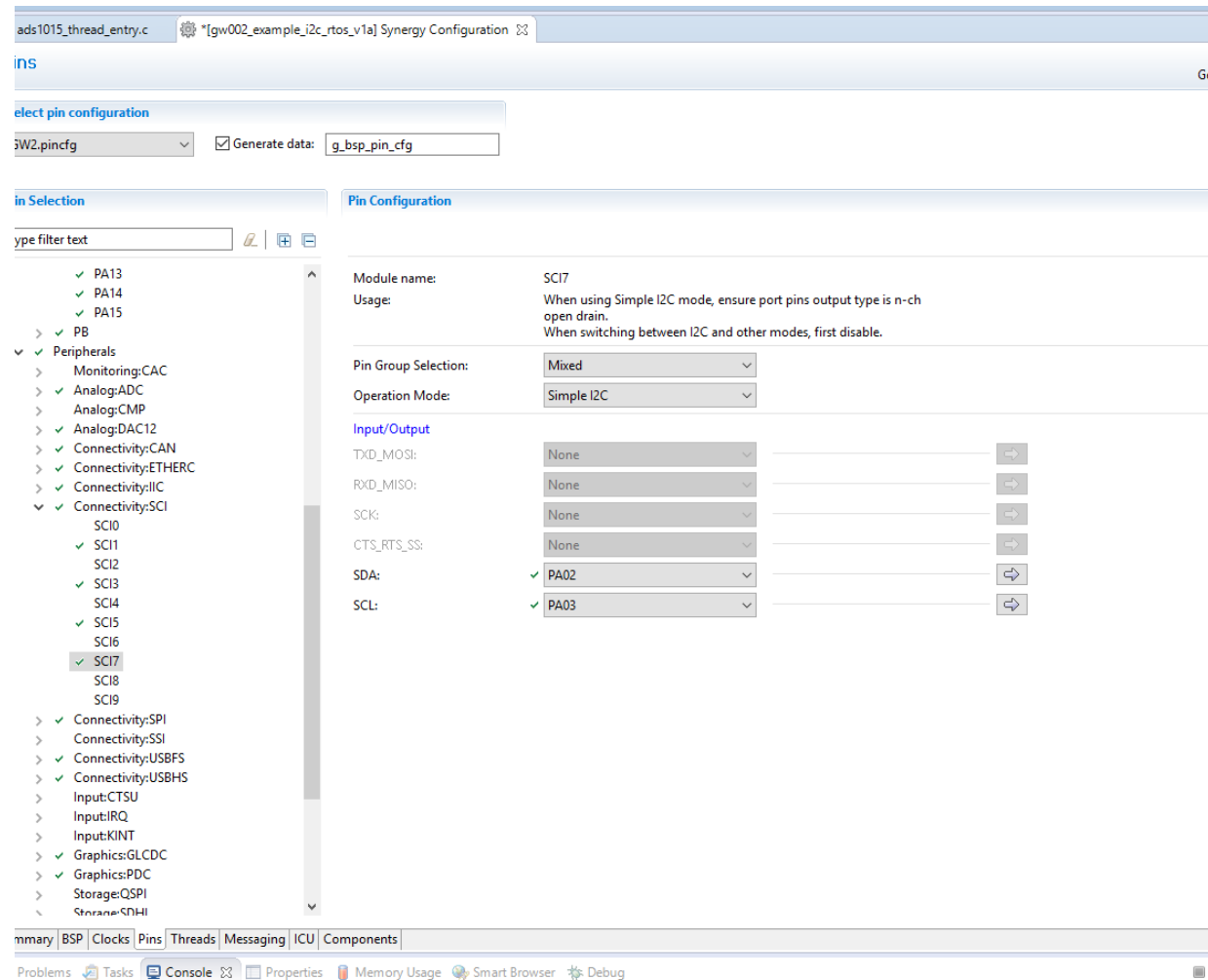
Table 8 – Pmod1 I2C Pin Connections

SCI7 Channel

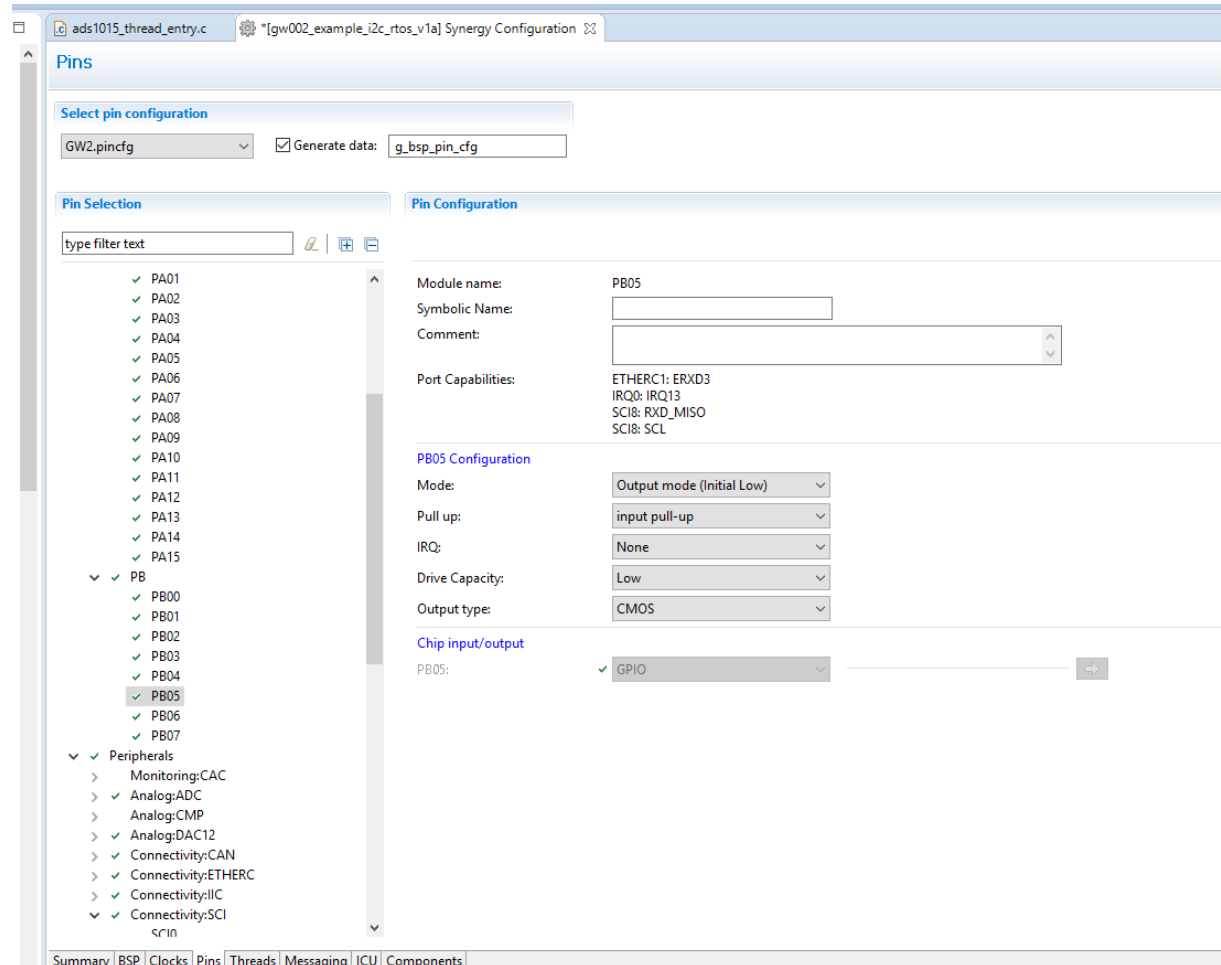
IO Expanders U18/U19 I2C port



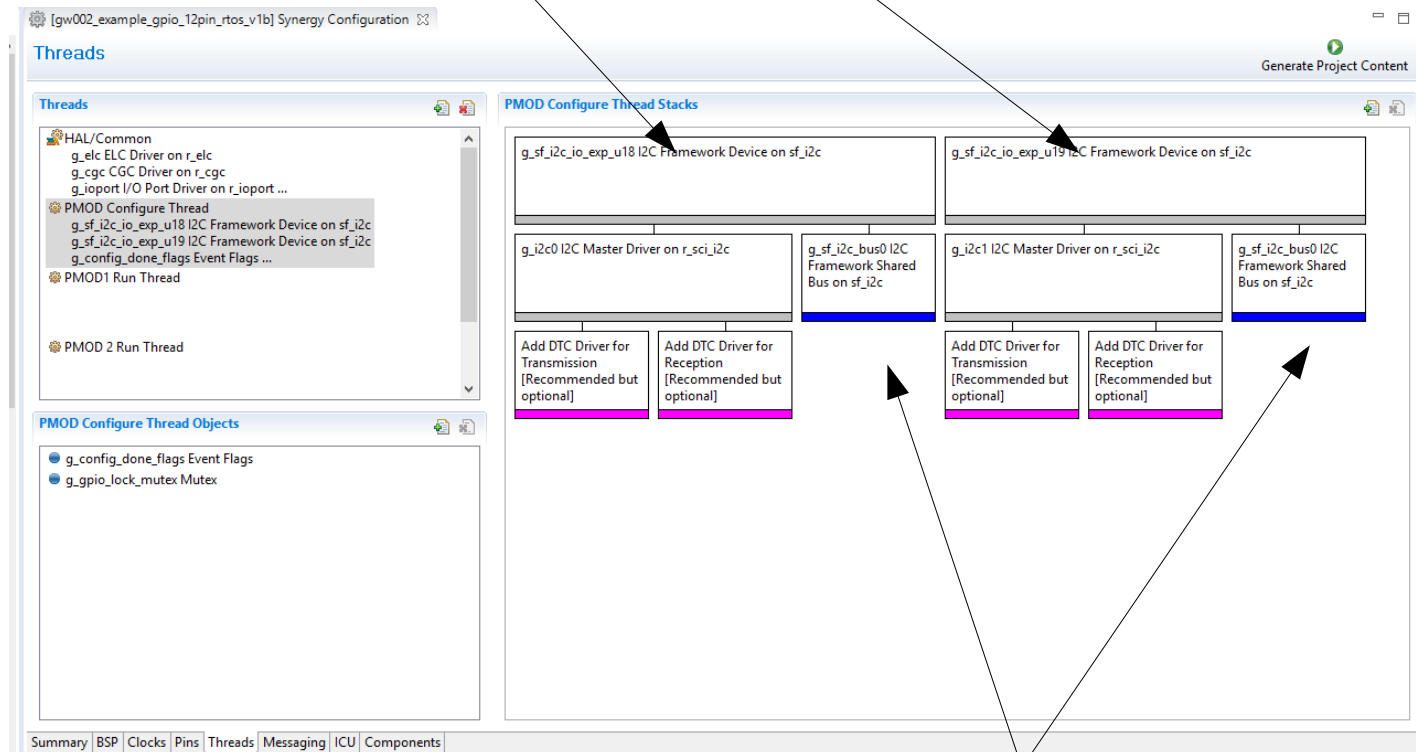
Configure PMOD 1 (Simple I2C)



PMOD 2 : PB02/3/4/5 Output Mode (P001 input only)



PMOD Configure Thread (Create U18/U19 I2C Framework Drivers)



Both drivers share the same g_sf_i2c_bus0 bus.

Properties

The PMOD Configure Thread Stacks window displays the following components:

- g_sf_i2c0 I2C Master Driver on r_sci_i2c**: Points to the **g_sf_i2c0 I2C Master Driver on r_sci_i2c** property window.
- g_sf_i2c1 I2C Master Driver on r_sci_i2c**: Points to the **g_sf_i2c1 I2C Master Driver on r_sci_i2c** property window.

g_sf_i2c0 I2C Master Driver on r_sci_i2c Properties:

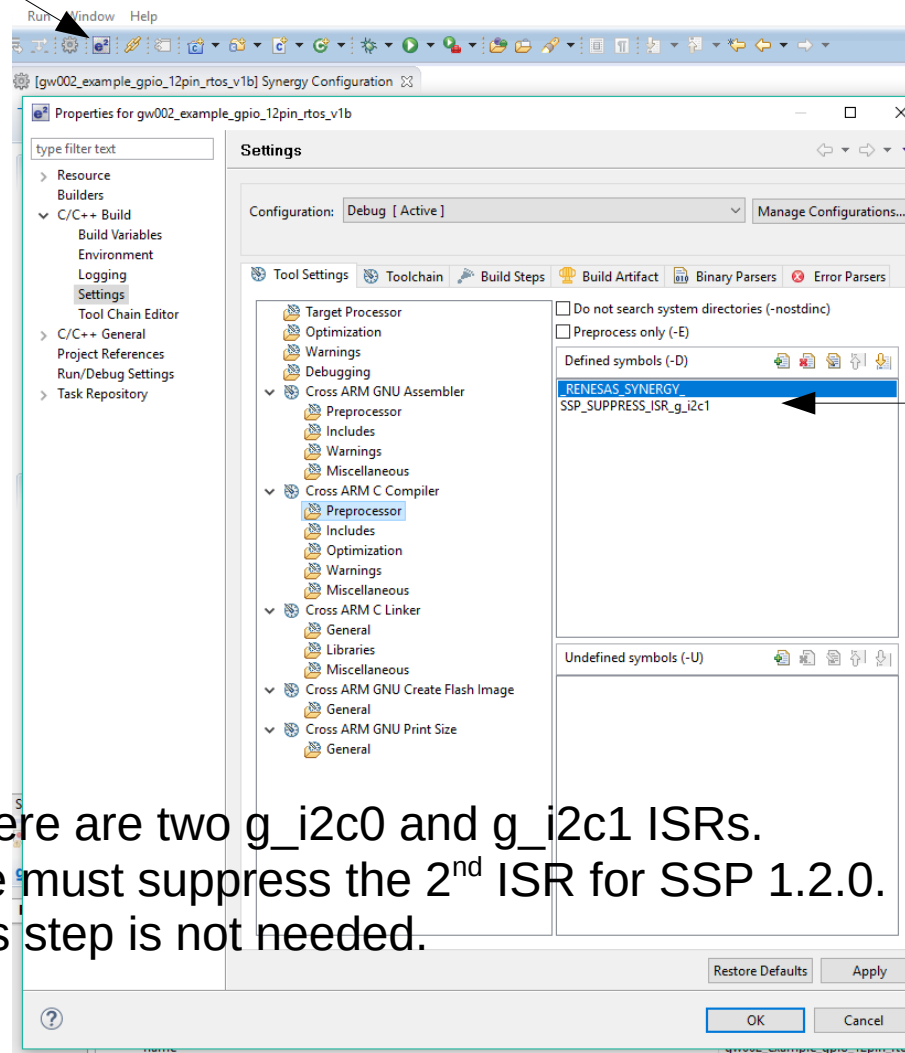
Property	Value
Module g_sf_i2c0 I2C Master Driver on r_sci_i2c	
Name	g_sf_i2c0
I2C Implementation	SCI I2C
Channel	3
Parameter Checking	Default (BSP)
Rate	Standard
Slave Address	0x25
Address Mode	7-Bit
SDA Output Delay (nano seconds)	300
Bit Rate Modulation Enable	Enable
Callback	NULL
Receive Interrupt Priority	Priority 2
Transmit Interrupt Priority	Priority 2
Transmit End Interrupt Priority	Priority 2

g_sf_i2c1 I2C Master Driver on r_sci_i2c Properties:

Property	Value
Module g_sf_i2c1 I2C Master Driver on r_sci_i2c	
Name	g_sf_i2c1
Channel	3
Rate	Standard
Slave Address	0x27
Address Mode	7-Bit
SDA Output Delay (nano seconds)	300
Bit Rate Modulation Enable	Enable
Callback	NULL
Receive Interrupt Priority	Priority 2
Transmit Interrupt Priority	Priority 2
Transmit End Interrupt Priority	Priority 2

Compiler Pre-processor Requirement

Click this icon!



Need to Suppress 2nd i2c ISR
(needed for SSP 1.2.0 only)

There are two g_i2c0 and g_i2c1 ISRs.
We must suppress the 2nd ISR for SSP 1.2.0. For later SSP versions,
this step is not needed.

PMOD 1 : I2C Framework Driver

The screenshot displays the I2C Framework Driver configuration in the PMOD 1 environment. The interface is divided into several panels:

- Threads:** Lists the threads running in the application, including 'PMOD Configure Thread', 'PMOD1 Run Thread', and 'PMOD2 Run Thread'.
- PMOD1 Run Thread Stacks:** A diagram showing the stack of threads. The 'g_sf_i2c_bus1 I2C Framework Shared Bus on sf_i2c' thread is highlighted, and an arrow points to its properties window.
- Properties:** A window showing the configuration for the 'g_sf_i2c_bus1 I2C Framework Shared Bus on sf_i2c'. The 'Information' tab is selected, showing the following properties:

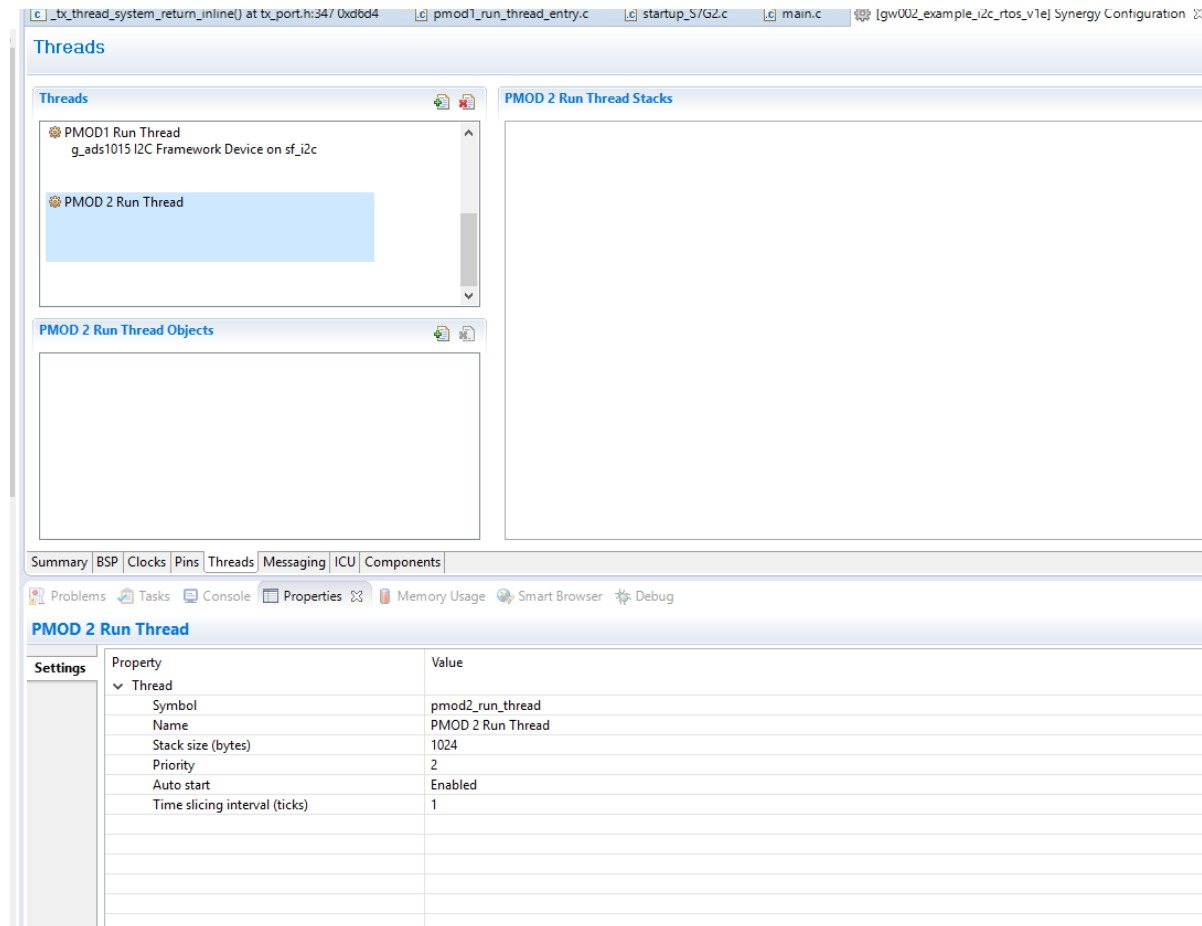
Property	Value
Name	g_sf_i2c_bus1
I2C Implementation	SCI I2C
Channel	7

Note: this i2c bus is not shared with U18/U19.

The bottom panel shows the 'g_i2c2 I2C Master Driver on r_sci_i2c' configuration. The 'Information' tab is selected, showing the following properties:

Property	Value
Name	g_i2c2
Channel	7
Rate	Fast-mode
Slave Address	0x48
Address Mode	7-Bit
SDA Output Delay (nano seconds)	300
Bit Rate Modulation Enable	Enable
Callback	NULL
Receive Interrupt Priority	Priority 2
Transmit Interrupt Priority	Priority 2
Transmit End Interrupt Priority	Priority 2

PMOD2 : No drivers needed for GPIO pins.



Configure PMOD 1 and PMOD 2 (pmod_configu_thread_entry.c)

```
*/
void setup_ioexp_registers(void) {

    //////////////////////////////////////
    /// U18 setup for PMOD1 : I2C 4PIN configuration
    //////////////////////////////////////
    pmod_bus_type_cfg[PMOD1_PORT] = I2C_TYPE_4PINS_COMH;

    ul8port0outreg.bit.pmod1_comms = set_pmod_com_bit(pmod_bus_type_cfg[PMOD1_PORT]);

    //////////////////////////////////////
    /// U19 setup for PMOD1 : Power enabled
    //////////////////////////////////////
    ul9port0outreg.bit.pmod1_power = 1; // pmod1 power enabled.

    //////////////////////////////////////
    /// U18 setup for PMOD2 : GPIO Type 1 12 pin configuration
    /// Set IO1-5 output pins (S7G2 MCU pins)
    /// Set IO6/7/8 output pins
    /// Set IO6/7/8 = 010
    ///
    /// Note IOS (This S7G2 pin is input mode only)
    //////////////////////////////////////
    pmod_bus_type_cfg[PMOD2_PORT] = GPIO_TYPE1_12PINS_COML;

    ul8port0outreg.bit.pmod2_comms = set_pmod_com_bit(pmod_bus_type_cfg[PMOD2_PORT]);

    ul8port0cfg.bit.pmod2_reset_io6 = SET_CFG_PIN_OUTPUT;
    ul8port1cfg.bit.pmod2_io7       = SET_CFG_PIN_OUTPUT;
    ul8port1cfg.bit.pmod2_io8       = SET_CFG_PIN_OUTPUT;

    ul8port0outreg.bit.pmod2_reset_io6 = 0; // make 0100 0000 initially
    ul8port1outreg.bit.pmod2_io7       = 0;
    ul8port1outreg.bit.pmod2_io8       = 0;

    //////////////////////////////////////
    /// U19 setup for PMOD2 : Power enabled
    //////////////////////////////////////
    ul9port0outreg.bit.pmod2_power = 1; // pmod2 power enabled.

}

/*-----*
 * End of File: pmod_configu_thread_entry.c
 *-----*/
```

PMOD 1 and 2 Threads' entry code

```
while (1) {
    // read configuration register for conversion status check
    ssp_err = ads1015_register_read (ADS1015_CONFIGREG_ADDR, &devicestatus);
    if (SSP_SUCCESS != ssp_err) {
        g_ioport.p_api->pinWrite(LEDREDPIN, LED_ON); // turn on red led for error
        while (1);
    }

    if (devicestatus & 0x8000) { // Conversion is ready
        g_ioport.p_api->pinWrite(LEDBLUEPIN, LED_ON); // turn on blue LED
        // read the conversion data
        ssp_err = ads1015_register_read (ADS1015_CONVERSION_ADDR, &convdata);
        if (SSP_SUCCESS != ssp_err) {
            g_ioport.p_api->pinWrite(LEDREDPIN, LED_ON); // turn on red led for error
        } else {
            convdata = convdata >> 4; // shift data down by 4 bits
            ain_level = (float)( convdata * 4.096 / 2048); // FSR = +/-4.096. For positive voltage range, only 2048 steps.
            // write configuration register and start the conversion! (Set for the continuous sampling mode)
            ssp_err = ads1015_register_write (ADS1015_CONFIGREG_ADDR, ADS1015_CONFIGREG_VALUE);
            if (SSP_SUCCESS != ssp_err) {
                g_ioport.p_api->pinWrite(LEDREDPIN, LED_ON); // turn on red led for error
                while (1);
            }
        }
    }
}
```

PMOD1: Start and Capture the ADC data.

```
#include "pmod2_run_thread.h"
#include "pmod_configure_thread_entry.h" // event flag
#include <pca9535/pca9535.h>

extern PMOD_BUS_TYPE_t pmod_bus_type_cfg[PMOD_PORT_NUM];

/* PMOD2 Run Thread entry function */
void pmod2_run_thread_entry(void)
{
    ULONG event_flags;
    uint8_t writedata[8] = {0x33, 0x4A, 0x33, 0xF1, 0x8F, 0x41, 0xCB, 0x99}; // write data

    tx_event_flags_get(&g_config_done_flags, IOEXP_DONE_EVENT_FLAG, TX_AND, &event_flags, TX_WAIT_FOREVER);

    while (true)
    {
        for (int i=0; i<8; i++) {
            tx_mutex_get(&g_gpio_lock_mutex, TX_WAIT_FOREVER);
            write_pmode_gpio_type1_byte_port (2, writedata[i], pmod_bus_type_cfg[PMOD2_PORT]);
            tx_mutex_put(&g_gpio_lock_mutex);
            tx_thread_sleep(10);
        }
    }
}
```

PMOD2 : Write data to the port.

Successful Build

The screenshot displays the Renesas e2 studio IDE interface. The main editor window shows the source code for `pm0d1_run_thread_entry.c`. The code includes headers for `pm0d1_run_thread.h`, `pm0d_configure_thread_entry.h`, and `pca9535/pca9535.h`. It defines a refresh delay of 10 ms and declares a global variable `g_pm0d2_port_data`. The `pm0d1_run_thread_entry` function is defined, which sets up event flags and enters a loop where it writes data to a port and sleeps for the refresh delay.

The Project Explorer on the left shows the project structure, with `gw002_example_i2c_rtos_v1e` selected. The Outline window on the right shows the project's structure, including the `pm0d1_run_thread_entry` function.

The Console window at the bottom shows the build output, indicating that the build was successful. The output includes the following text:

```
CDT Build Console [gw002_example_i2c_rtos_v1e]
'Finished building target: gw002_example_i2c_rtos_v1e.elf'
'Invoking: Cross ARM GNU Create Flash Image'
'Invoking: Cross ARM GNU Print Size'
arm-none-eabi-objcopy -O srec "gw002_example_i2c_rtos_v1e.elf" "gw002_example_i2c_rtos_v1e.srec"
arm-none-eabi-size --format=berkeley "gw002_example_i2c_rtos_v1e.elf"
text data bss dec hex filename
51664 356 13012 65032 fe08 gw002_example_i2c_rtos_v1e.elf
'Finished building: gw002_example_i2c_rtos_v1e.srec'
'Finished building: gw002_example_i2c_rtos_v1e.siz'
14:13:18 Build Finished. 0 errors, 3 warnings. (took 8s.284ms)
```

The status bar at the bottom indicates the project name `gw002_example_i2c_rtos_v1e`.

Run in Debug Mode: Place a break point to see the ADC result.

Debug - gw002_example_i2c_rtos_v1e/src/pmod1_run_thread_entry.c - e2 studio

File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help

Debug

gw002_example_i2c_rtos_v1e Debug [Renesas GDB Hardware Debugging]

gw002_example_i2c_rtos_v1e.elf [1]

Thread #1 1 (single core) (Suspended : Breakpoint)

pmod1_run_thread_entry() at pmod1_run_thread_entry.c:63 0x8f3a

pmod1_run_thread_func() at pmod1_run_thread.c:114 0x7dba

_tx_thread_shell_entry() at tx_thread_shell_entry.c:164 0xa384

0xffffffff

Thread #2 1001 (single core - PMOD Configure Thread [Sleeping RC:76]) (Suspended : Signal : SIGTRAP:Trace/breakpoint trap)

_tx_thread_system_return_inline() at tx_port.h:347 0xa584

_tx_thread_system_suspend() at tx_thread_system_suspend.c:710 0xa584

_tx_thread_sleep() at tx_thread_sleep.c:215 0xa40a

pmod_configure_thread_entry() at pmod_configure_thread_entry.c:127 0x9090

pmod_configure_thread_func() at pmod_configure_thread.c:160 0x7f7e

(x) Variables

Breakpoints Registers Modules Expressions Eventpoints IO Registers Peripherals

Name	Type	Value
event_flags	ULONG	1
ssp_err	ssp_err_t	SSP_SUCCESS
deviceconfig	uint16_t	17283
devicestatus	uint16_t	50051
convdata	uint16_t	540
ain_level	float	1.08000004

_tx_thread_system_return_inline() at tx...

pmod1_run_thread_entry.c

```
53 00008eba if (devicestatus & 0x8000) { // Conversion is ready
54 00008ec4 q_ioport.p_api->pinWrite(LED_BLUEPIN, LED_ON); // turn on blue LED
55 // read the conversion data
56 00008ed2 ssp_err = ads1015_register_read (ADS1015_CONVERSION_ADDR, &convdata);
57 00008ee0 if (SSP_SUCCESS != ssp_err) {
58 00008ee6 q_ioport.p_api->pinWrite(LED_RED_PIN, LED_ON); // turn on red led for error
59 } else {
60 00008ef6 convdata = convdata >> 4; // shift data down by 4 bits
61 00008efe ain_level = (float) (convdata * 4.096 / 2048); // FSR = +/-4.096. For positive voltage range, only 2048 steps.
62 // write configuration register and start the conversion! (Set for the continuous sampling mode)
63 00008f3a ssp_err = ads1015_register_write (ADS1015_CONFIGREG_ADDR, ADS1015_CONFIGREG_VALUE);
64 00008f48 if (SSP_SUCCESS != ssp_err) {
65 00008f4e q_ioport.p_api->pinWrite(LED_RED_PIN, LED_ON); // turn on red led for error
66 00008f5c while (1);
67 }
68 }
69 }
70 00008f5e tx_thread_sleep (10); // 100ms
71 00008f64 }
72 }
73 }
74 }
```

ads1015

ADS1015.c

ADS1015.h

pca9535

synergy_gen

hal_entry.c

pmod_configure_thread_entry.c

pmod_configure_thread_func.c

Console

Tasks Renesas Coverage Memory Usage Performance Analysis Profile Real-time Chart Trace Visual Expression ARM CoreSight ITM Live Trace Console Smart Browser Problems Executables Memory

gw002_example_i2c_rtos_v1e Debug [Renesas GDB Hardware Debugging] C:/Renesas/e2_studio531_s7g2avnet/DebugComp/arm-none-eabi-gdb (7.8.2)

63 ssp_err = ads1015_register_write (ADS1015_CONFIGREG_ADDR, ADS1015_CONFIGREG_VALUE);

Program received signal SIGTRAP, Trace/breakpoint trap.

_tx_thread_system_return_inline () at inc\el1/.cm4_gcc/tx_port.h:347

347 inc\el1/.cm4_gcc/tx_port.h: No such file or directory.

Breakpoint 2, pmod1_run_thread_entry () at ../src/pmod1_run_thread_entry.c:63

63 ssp_err = ads1015_register_write (ADS1015_CONFIGREG_ADDR, ADS1015_CONFIGREG_VALUE);

Program received signal SIGTRAP, Trace/breakpoint trap.

_tx_thread_system_return_inline () at inc\el1/.cm4_gcc/tx_port.h:347

347 inc\el1/.cm4_gcc/tx_port.h: No such file or directory.

Suspended

Writable Smart Insert 6:20

Analog Scope Result (A slight difference)

