

# Avnet Visible Things UART 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	Type 2	Type 2A	Type 3	Type 4	Type 4A	Type 5	Type 6
	I2C	GPIO	SPI	Expanded SPI	UART	UART	Expanded UART	H- Bridge	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

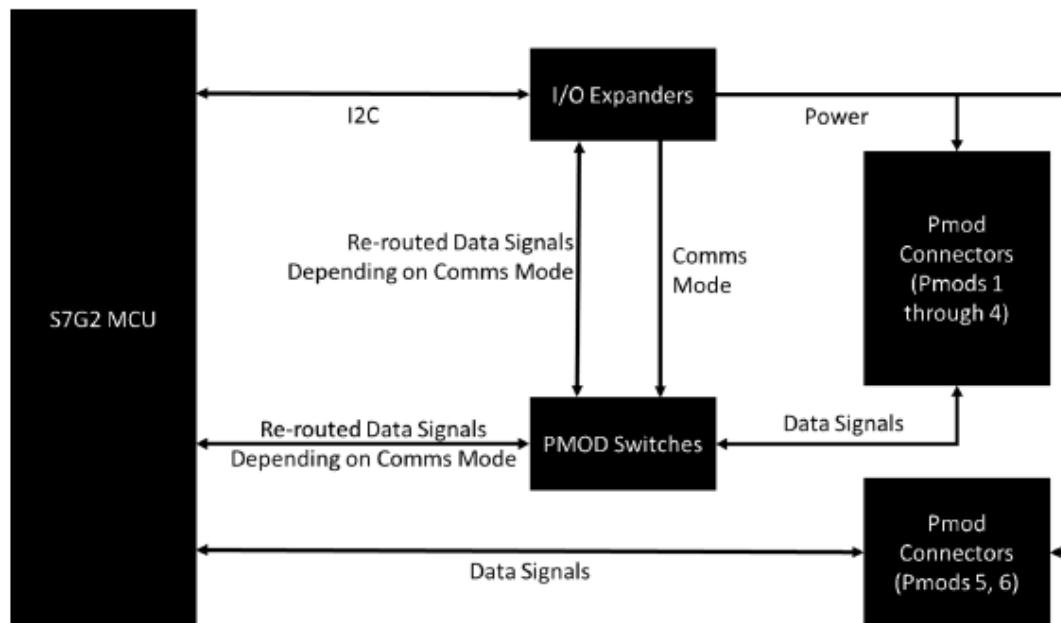
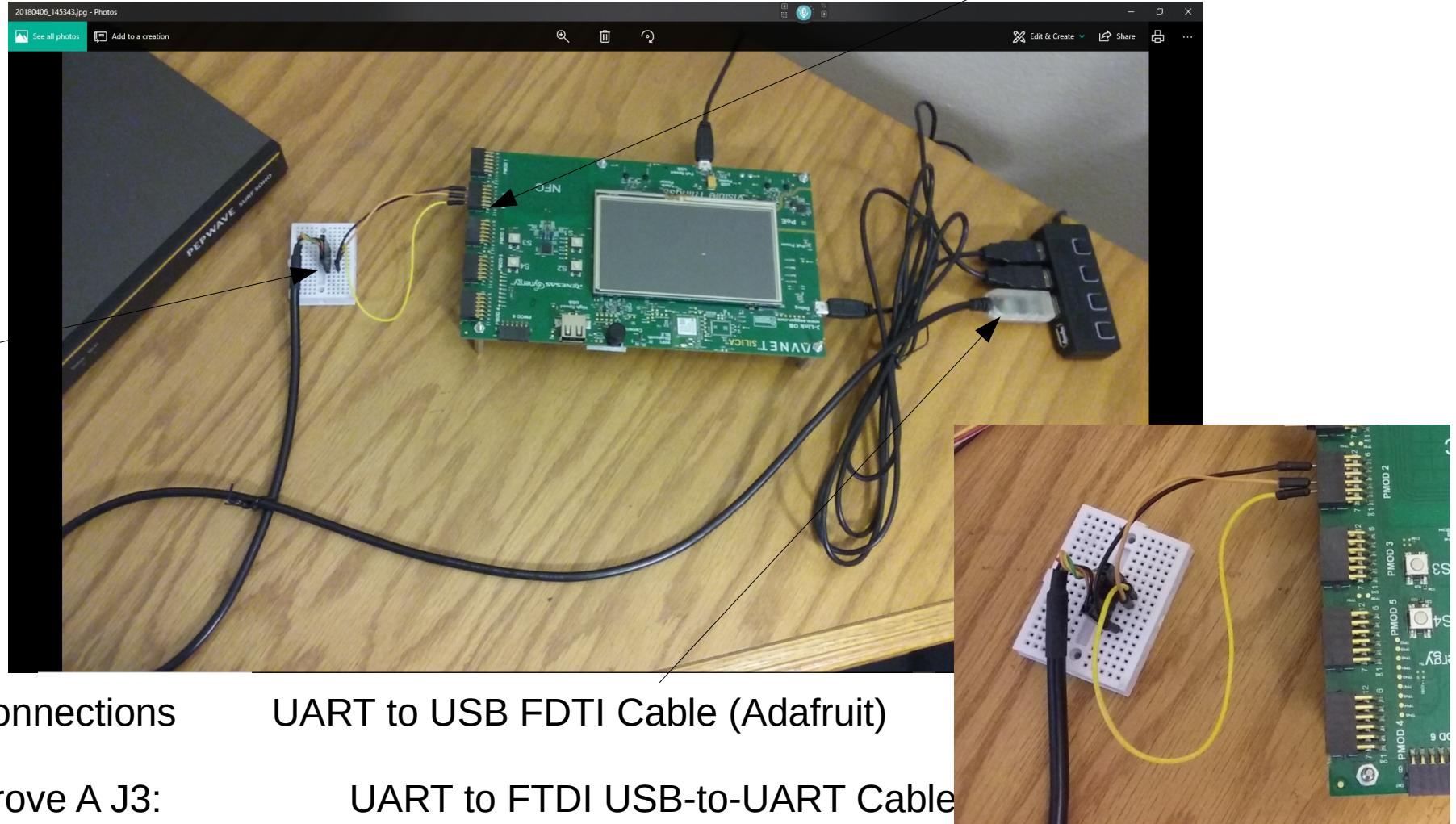


Figure 30 – Pmod, I/O Expander Interconnections Diagram

# Hardware Setup



## Connections

## UART to USB FTDI Cable (Adafruit)

## Grove A J3:

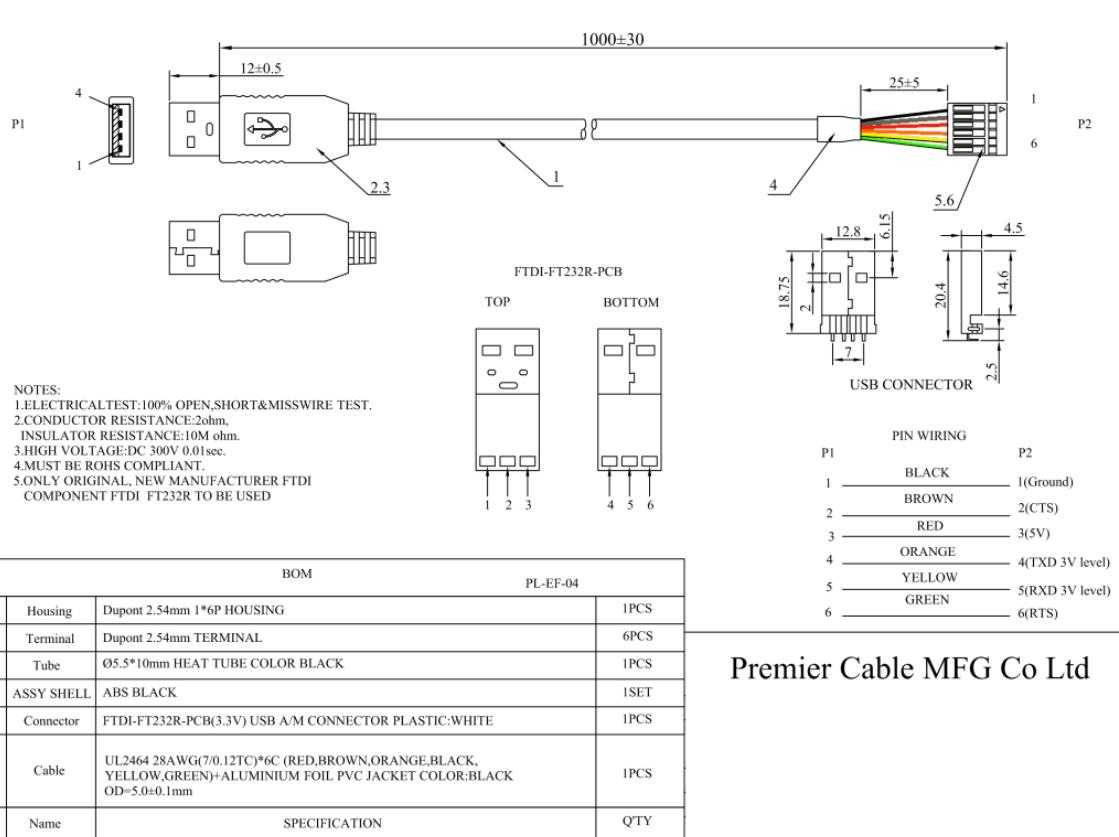
# UART to FTDI USB-to-UART Cable

Pin 2: S2G7 PA02 TXD  
Pin 3: S2G7 PA03 RXD  
Pin 5: GND

- Pin 5 (Yellow) RXD
- Pin 4 (Orange) TXD
- Pin 1 (Black) GND

# FTDI Serial TTL-232 USB Cable

## PRODUCT ID: 70



**The version we have is the 3.3V.** The data signals are at 3V and the power line provides 5V. We suggest this for any product that needs FTDI cables. Because the cable is 5V-logic compliant, **you can use it with 3v or 5v logic just fine - no level shifting required!**

If you have a device that is running at 5V logic and requires 5V power, this cable will work fine. If you want to tweak the voltages and signals a little, you should also check out the FTDI friend

# PMOD 1 Port

## GPIO Type 1 Pmods

If clearance is not an issue, two GPIO Type 1 standard Pmods can be used together when the connector is configured in UART/SPI Comms Mode. **Do not turn a second Type 1 Pmod upside down to get it to fit! Damage may occur as a result of inserting a Pmod upside down!**

	Pmod		S7G2	
	Pin	Pin Function	Port Pin	Pin Function
Upper Row	1	GPIO	PA05	GPIO
	2	GPIO	PA02	GPIO
	3	GPIO	PA03	GPIO
	4	GPIO	PA04	GPIO
	5	GND	GND	GND
	6	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)
Lower Row	1	GPIO	P400	GPIO
	2	GPIO	I/O Expander	GPIO
	3	GPIO	I/O Expander	GPIO
	4	GPIO	I/O Expander	GPIO
	5	GND	GND	GND
	6	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)

Table 7 – Pmod1 GPIO Type 1 Pin Connections (UART/SPI Mode)

# PMOD 2 Port

## UART/SPI Comms Mode

### UART Type 4 and SPI Type 2 Pmods

UART Type 4 Pmods and SPI Type 2 Pmods only have 6 pins instead of the 12 provided by the Gateway's connector. Insert these Pmods into the upper row of the connector (connector pins 1 through 6).

Pmod			S7G2		
Pin	Pin Function by Type		Port Pin	Pin Function	
	UART (4, 4A)	SPI (2, 2A)		UART	SPI
1	RTS	CS	PB02	CTS	CTS_RTS_SS
2	RX	MISO	PB04	TXD_MOSI	TXD_MOSI
3	TX	MOSI	PB05	RXD_MISO	RXD_MISO
4	CTS	SCK	PB03	RTS	SCK
5	GND	GND	GND	GND	GND
6	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)

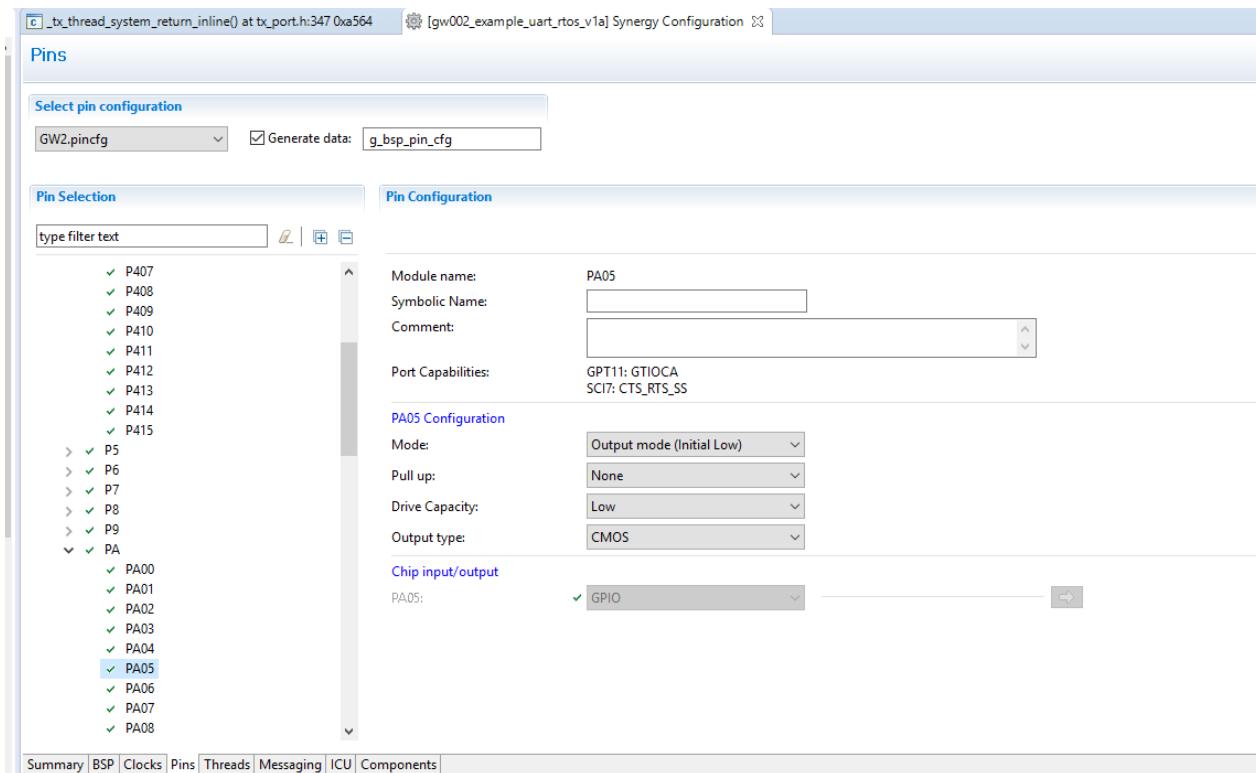
Table 12 – Pmod2 UART Type 4 and SPI Type 2 Pin Connections



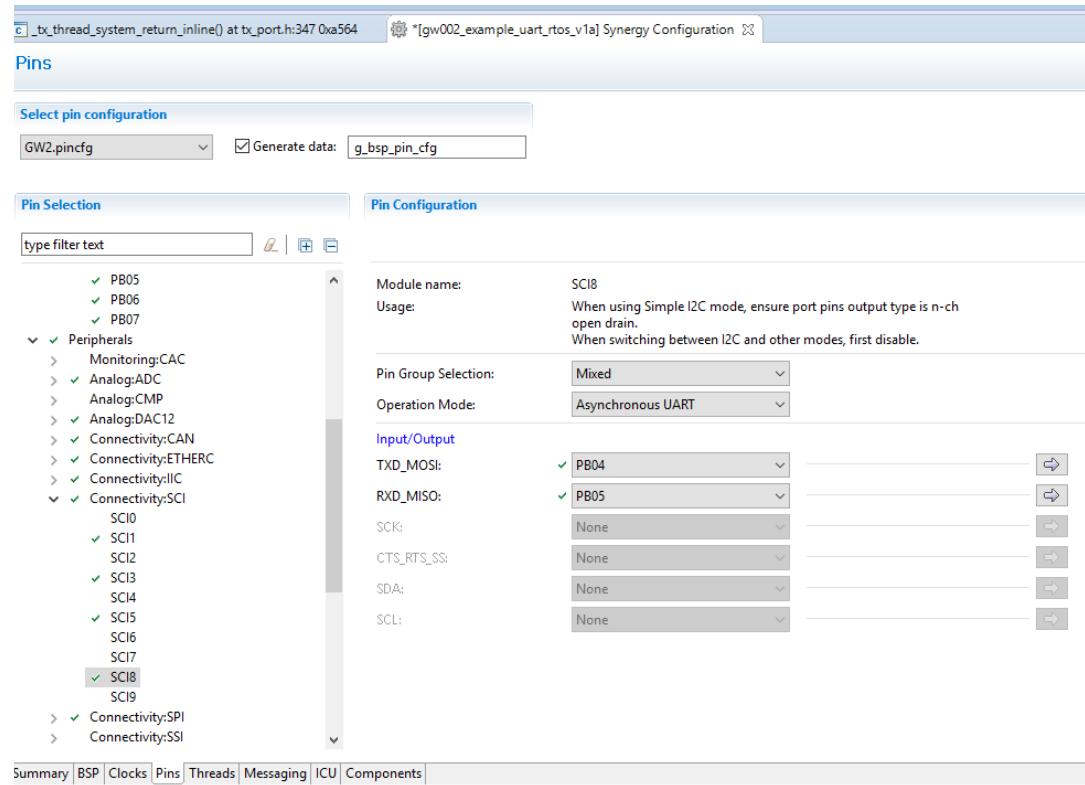
# IO Expanders U18/U19 I2C port

The screenshot shows the Synergy Configuration software interface for a project titled "[gw02\_example\_i2c\_rtos\_v1e]". The main window is titled "Pins" and displays the "Select pin configuration" dialog. The configuration is set to "GW2.pincfg" and has the option "Generate data: g\_bsp\_pin\_cfg" checked. The "Pin Selection" panel on the left lists various connectivity modules: Analog: CMP, Analog: DAC12, Connectivity: CAN, Connectivity: ETHERC, Connectivity: IIC, and Connectivity: SCI. Under Connectivity: SCI, several pins are listed: SCI0, SCI1, SCI2, SCI3, SCI4, SCI5, SCI6, SCI7, SCI8, SCI9, Connectivity: SPI, Connectivity: SSI, Connectivity: USBFS, Connectivity: USBHS, Input: CTSU, Input: IRQ, and Input: KINT. The "Pin Configuration" panel on the right shows the configuration for SCI3. It includes fields for "Module name: SCI3", "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: Simple I2C", and "Input/Output" settings for TXD\_MOSI, RXD\_MISO, SCK, CTS\_RTS\_SS, SDA, and SCL. The SDA field is set to "P707" and the SCL field is set to "P706".

# PMOD 1 : PA02/3/4/5 Output Mode (P400 output only)

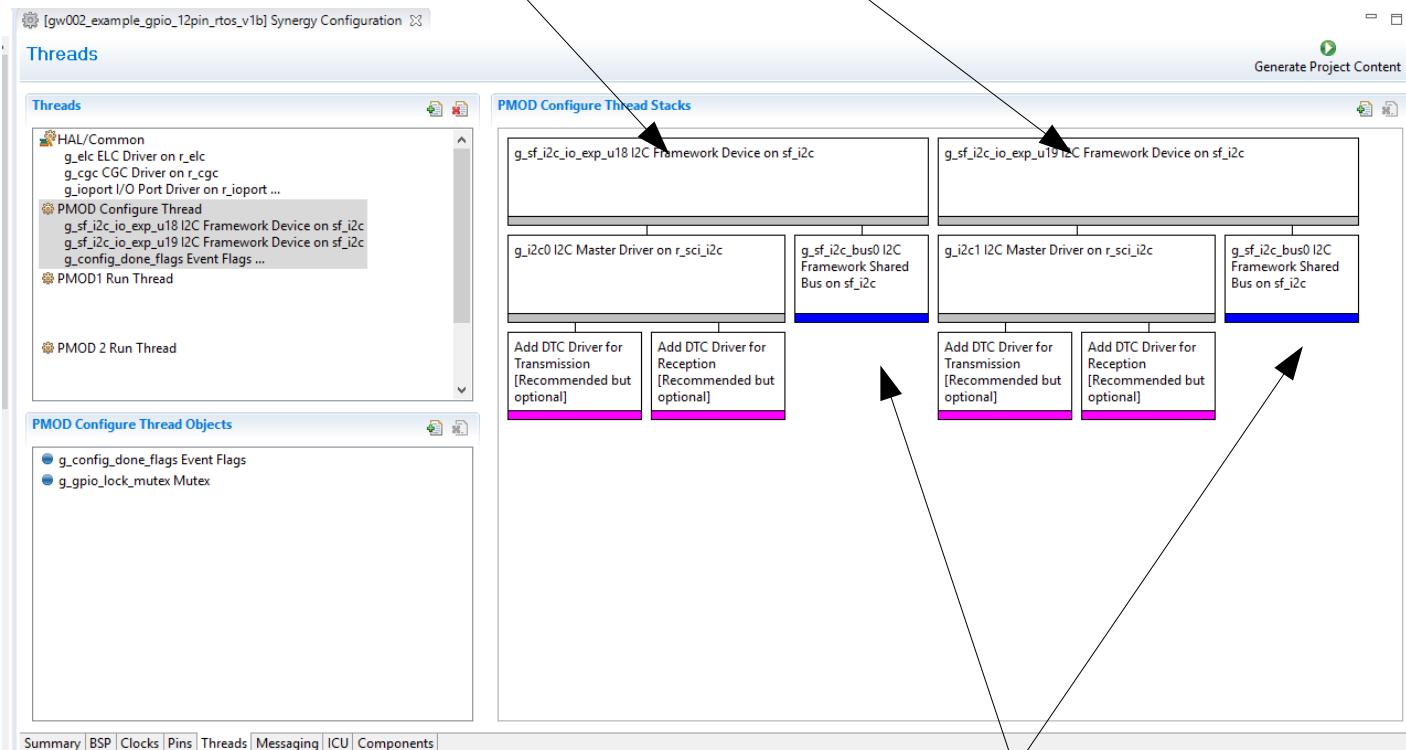


# Configure PMOD 2 (Asy UART)





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



Both drivers share the same g\_sf\_i2c\_bus0 bus.

# Properties

**PMOD Configure Thread Stacks**

g_sf_i2c_io_exp_u18 I2C Framework Device on sf_i2c	g_sf_i2c_io_exp_u19 I2C Framework Device on sf_i2c
g_i2c0 I2C Master Driver on r_sci_i2c	g_sf_i2c_bus0 I2C Framework Shared Bus on sf_i2c
g_i2c1 I2C Master Driver on r_sci_i2c	g_sf_i2c_bus0 I2C Framework Shared Bus on sf_i2c

**g\_sf\_i2c\_bus0 I2C Framework Shared Bus on sf\_i2c**

Property

Name	g_sf_i2c_bus0
I2C Implementation	SCI I2C
Channel	3

Value

- Add DTC Driver for Transmission [Recommended but optional]
- Add DTC Driver for Reception [Recommended but optional]

**g\_i2c0 I2C Master Driver on r\_sci\_i2c**

Property

Name	g_i2c0
Channel	3
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\_i2c1 I2C Master Driver on r\_sci\_i2c**

Property

Name	g_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

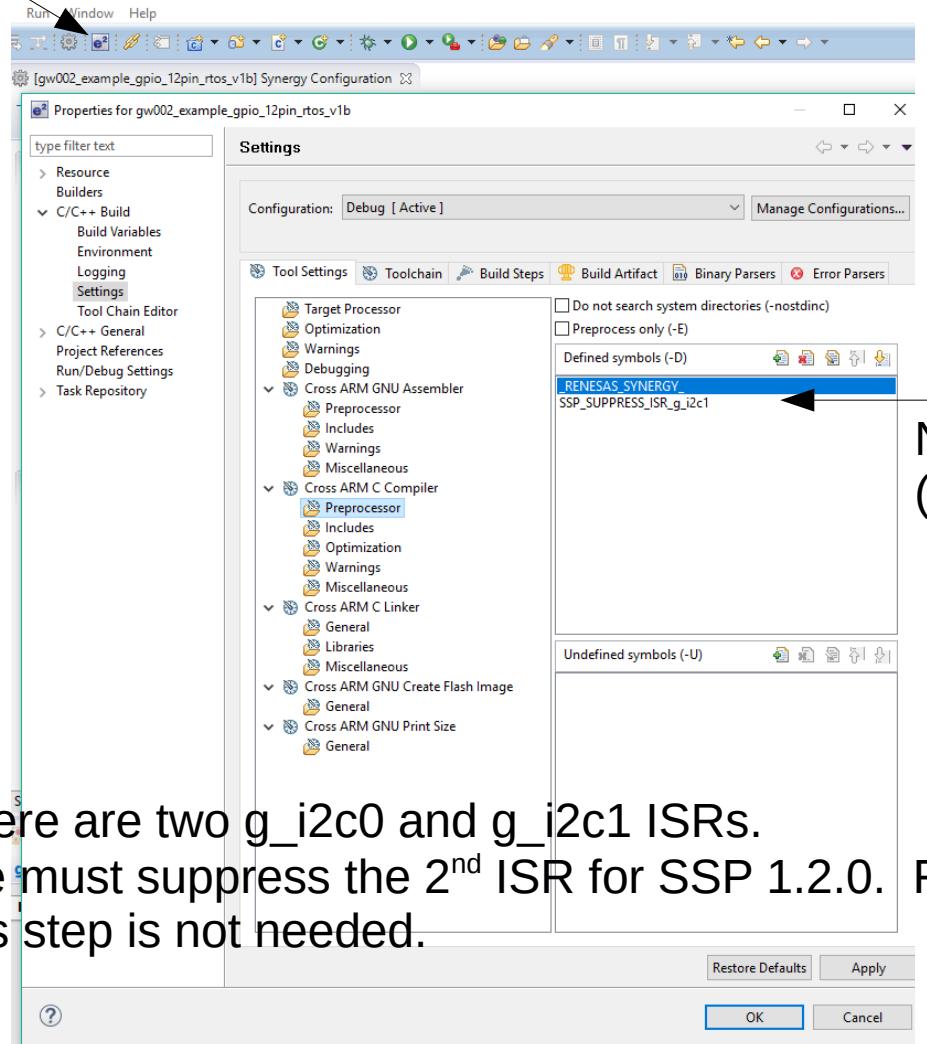
Smart Browser    Debug

Default (BSP)

I2C by Michael Li

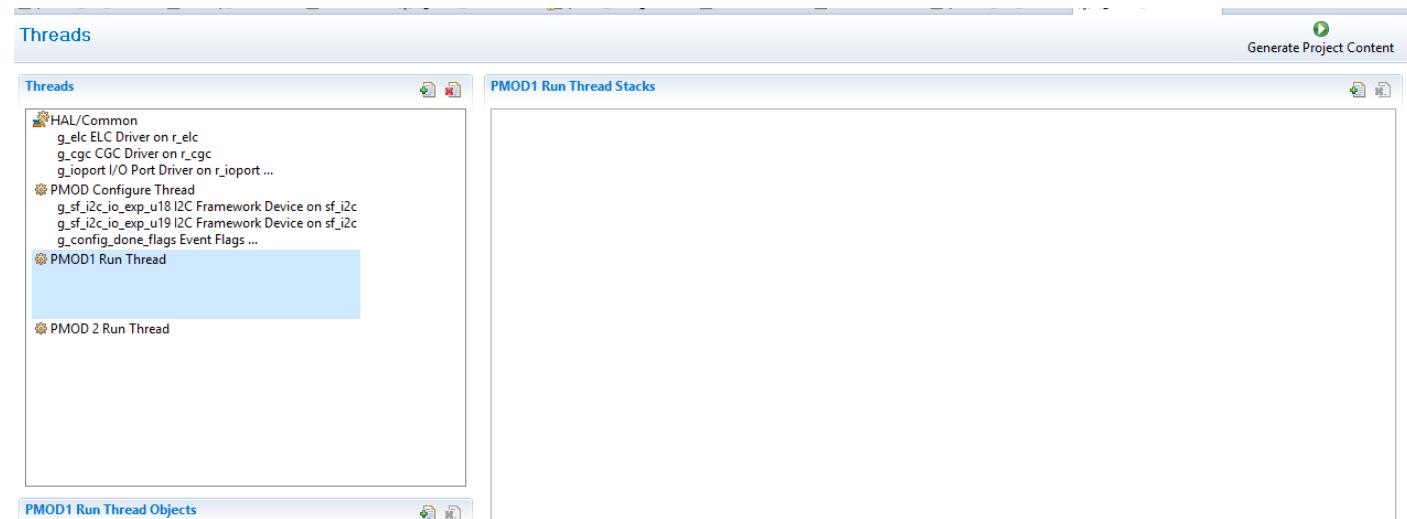
# Compiler Pre-processor Requirement

Click this icon!

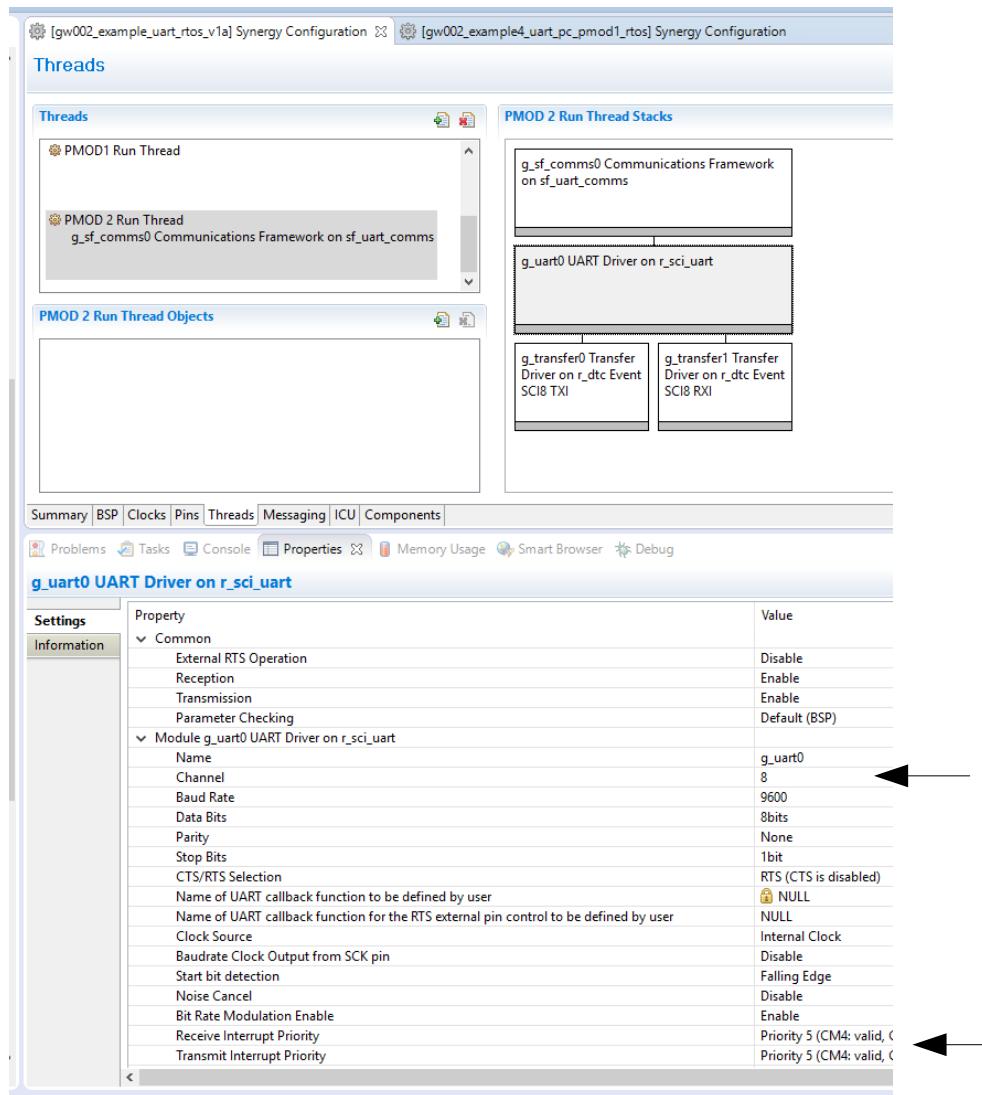


There are two g\_i2c0 and g\_i2c1 ISRs.  
We must suppress the 2<sup>nd</sup> ISR for SSP 1.2.0. For later SSP versions,  
this step is not needed.

PMOD 1 Thread has no drivers needed because the port is configured as a GPIO port.



# PMOD 2 Run Thread : UART Framework Driver





# Configure PMOD 1 and PMOD 2

```
uart_rtos_v1c] Synergy Configuration [pmod_configure_thread_entry.c] [x]
/*
 * Return : None
 *
 */
void setup_ioexp_registers(void) {

    //////////////////////////////////////////////////////////////////
    /// U18 setup for PMOD1 : GPIO Type 1 12 pin configuration
    ///                         Set IO1-5 output pins (S7G2 MCU pins)
    ///                         Set IO6/7/8 output pins
    ///                         Set IO6/7/8 =001
    //////////////////////////////////////////////////////////////////
    pmod_bus_type_cfg[PMOD1_PORT] = GPIO_TYPE1_12PINS_COML;

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

    ul8port0cfg.bit.pmod1_reset_io6 = SET_CFG_PIN_OUTPUT;
    ul8port1cfg.bit.pmod1_io7      = SET_CFG_PIN_OUTPUT;
    ul8port1cfg.bit.pmod1_io8      = SET_CFG_PIN_OUTPUT;

    ul8port0outreg.bit.pmod1_reset_io6 = 1; // make 0010 0000 initially
    ul8port1outreg.bit.pmod1_io7     = 0;
    ul8port1outreg.bit.pmod1_io8     = 0;

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

    //////////////////////////////////////////////////////////////////
    /// U18 setup for PMOD2 : UART 4 6 pin configuration
    //////////////////////////////////////////////////////////////////
    pmod_bus_type_cfg[PMOD2_PORT] = UART_TYPE4_6PINS_COML;

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

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

}

/*
 *-----*
 * End of File: pmod_configure_thread_entry.c
 *-----*/

```

# PMOD 1 Thread Entry

```
rt_rtos_v1c] Synergy Configuration  pmod_configure_thread_entry.c  pmod1_run_thread_entry.c  pmod2_run_thread_entry.c
#include "pmod1_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];

/* PMOD1 Run Thread entry function */
void pmod1_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); // Don't clear it. Leave it

    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 (1, writedata[i],pmod_bus_type_cfg[PMOD1_PORT]);
            tx_mutex_put(&g_gpio_lock_mutex);
            tx_thread_sleep(10);
        }
    }
}
```

# PMOD 2 Thread Entry

```
0000baf0    tx_event_flags_get(&g_config_done_flags, IOEXP_DONE_EVENT_FLAG, TX_AND, &event_flags, TX_WAIT_FOREVER); // Don't clear it. Leave it e
0000baf1    // SSP_ERR_IN_USE: Maybe it is already open initially
0000baf2    //ReturnVal = g_sf_comms0.p_api->open (g_sf_comms0.p_ctrl, g_sf_comms0.p_cfg);
0000baf3    //if (SSP_SUCCESS != ReturnVal)
0000baf4    //{
0000baf5        // g_ioport.p_api->pinWrite(LEDREDPIN, IOPORT_LEVEL_HIGH);
0000baf6        // while(1);
0000baf7    //}

0000bb12    // Demonstrate the sprintf can handle floating point value. Enable Floating support in the compiler option.
0000bb13    sprintf(g_buffer, "Type %d characters, they will be echoed back. (Board VCC = %5.2f V\r\n", RECEIVE_STRING_LENGTH, 3.30);
0000bb14    ReturnVal = g_sf_comms0.p_api->write(g_sf_comms0.p_ctrl, (unsigned char *)g_buffer, strlen(g_buffer), TX_WAIT_FOREVER);
0000bb15    if (SSP_SUCCESS != ReturnVal)
0000bb16    {
0000bb17        g_ioport.p_api->pinWrite(LEDREDPIN, IOPORT_LEVEL_HIGH);
0000bb18        while(1);
0000bb19    }

0000bb5e    while (1)
0000bb60    {
0000bb61        memset(g_buffer, 0, sizeof(g_buffer));

0000bb68    ReturnVal = g_sf_comms0.p_api->read(g_sf_comms0.p_ctrl, (unsigned char *)g_buffer, RECEIVE_STRING_LENGTH, TX_WAIT_FOREVER);
0000bb70    if (SSP_SUCCESS != ReturnVal)
0000bb71    {
0000bb72        g_ioport.p_api->pinWrite(LEDREDPIN, IOPORT_LEVEL_HIGH);
0000bb73        while(1);
0000bb74    }

0000bb98    if (l3 == g_buffer[0])
0000bb99    sprintf(g_buffer + RECEIVE_STRING_LENGTH, "\r\n"); // append to the end of the location designated by g_buffer
0000bb9a    // + RECEIVE_STRING_LENGTH

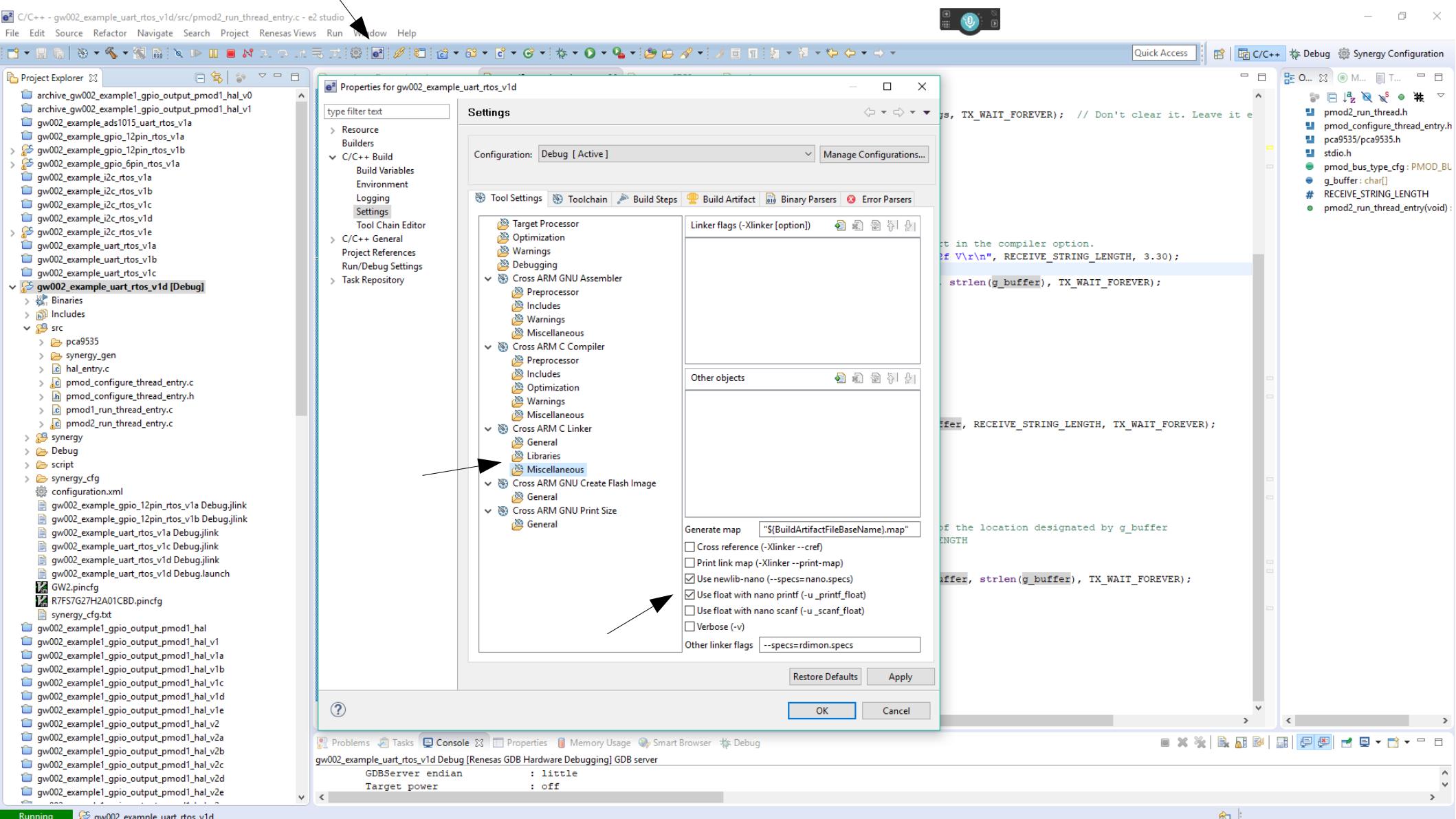
0000bbae    ReturnVal = g_sf_comms0.p_api->write(g_sf_comms0.p_ctrl, (unsigned char *)g_buffer, strlen(g_buffer), TX_WAIT_FOREVER);
0000bbbd    if (SSP_SUCCESS != ReturnVal)
0000bbbf    {
0000bbbf        while(1);
0000bbbf        tx_thread_sleep (1);
0000bbbf    }
0000bbde    }
```

Print a message with an integer value and a decimal value.

Echo every typed character.  
When a return character is detected, send a newline character.

# Enable floating point support for printf

Click this icon





C/C++ - gw002\_example\_gpio\_i2c\_spi\_uart\_v1h/src/pmod1\_run\_thread\_entry.c - e2 studio

File Edit Source Refactor Navigate Project Renesas Views Run Window Help

Project Explorer

- gw002\_example\_gpio\_12pin\_rtos\_v1b
- gw002\_example\_gpio\_6pin\_rtos\_v1a
- gw002\_example\_gpio\_i2c\_spi\_uart\_v1h
- gw002\_example\_i2c\_rtos\_v1e
- gw002\_example\_spi\_tos\_v1a
- gw002\_example\_uart\_rtos\_v1d [Debug]**
  - Binaries
  - Includes
  - src
  - synergy
  - Debug
  - script
  - synergy\_cfg
  - configuration.xml
  - gw002\_example\_gpio\_12pin\_rtos\_v1a Debug.jlink
  - gw002\_example\_gpio\_12pin\_rtos\_v1b Debug.jlink
  - gw002\_example\_uart\_rtos\_v1a Debug.jlink
  - gw002\_example\_uart\_rtos\_v1c Debug.jlink
  - gw002\_example\_uart\_rtos\_v1d Debug.jlink
  - gw002\_example\_uart\_rtos\_v1d Debug.launch
  - GW2.pincfg
  - R7F57G27H2A01CBD.pincfg
  - synergy\_cfg.txt

Outline

```

pmod1_run_thread.h
pmod_configure_thread_entry.h // event flag
pca9535/pca9535.h

extern PMOD_BUS_TYPE_t pmod_bus_type_cfg[PMOD_PORT_NUM];

#define refreshleddelay 10 // 10x10 msec delay time.

extern uint8_t g_pmod2_port_data;

/* PMOD1 Run Thread entry function */
void pmod1_run_thread_entry(void)
{
    ULONG event_flags;
    //uint8_t writedata[8] = {0xAA, 0x00, 0x55, 0x00, 0x55, 0x00, 0xAA, 0x00}; // write data
    uint8_t writedata[8] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80}; // 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_typep_port (1, -g_pmod2_port_data,pmod_bus_type_cfg[PMOD1_PORT]);
            tx_mutex_put(&g_gpio_lock_mutex);
            tx_thread_sleep(refreshleddelay);
        }
    }
}

```

Problems Tasks Console Properties Memory Usage Smart Browser

CDT Build Console [gw002\_example\_uart\_rtos\_v1d]

```

'Finished building target: gw002_example_uart_rtos_v1d.elf'
'
'Invoking: Cross ARM GNU Create Flash Image'
arm-none-eabi-objcopy -O srec "gw002_example_uart_rtos_v1d.elf" "gw002_example_uart_rtos_v1d.srec"
'Invoking: Cross ARM GNU Print Size'
arm-none-eabi-size --format=berkeley "gw002_example_uart_rtos_v1d.elf"
text    data    bss   dec   hex filename
73828    904   13296   88028   157de gw002_example_uart_rtos_v1d.elf
'Finished building: gw002_example_uart_rtos_v1d.siz'
'Finished building: gw002_example_uart_rtos_v1d.srec'
'
'

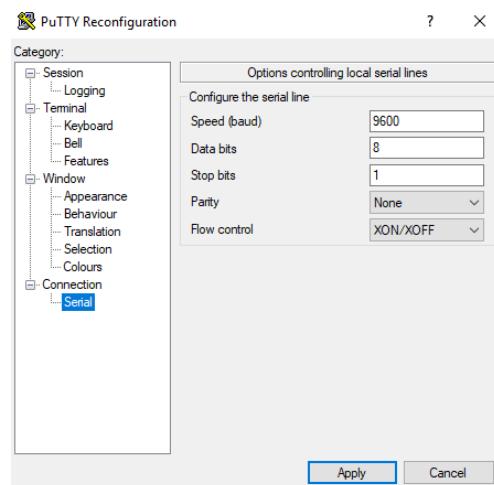
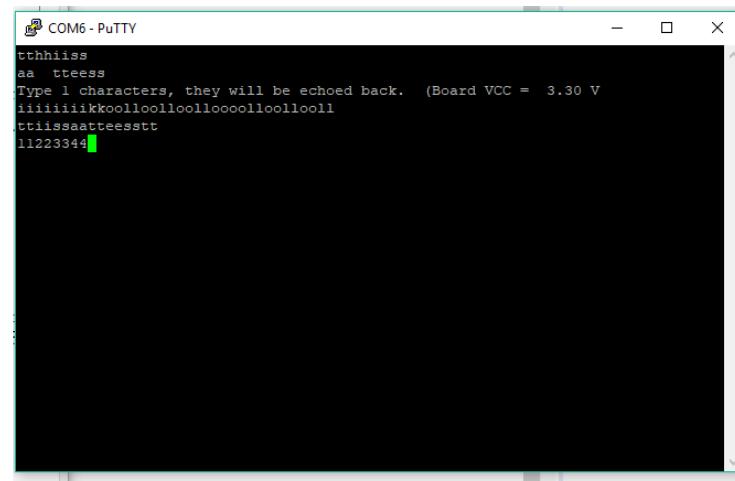
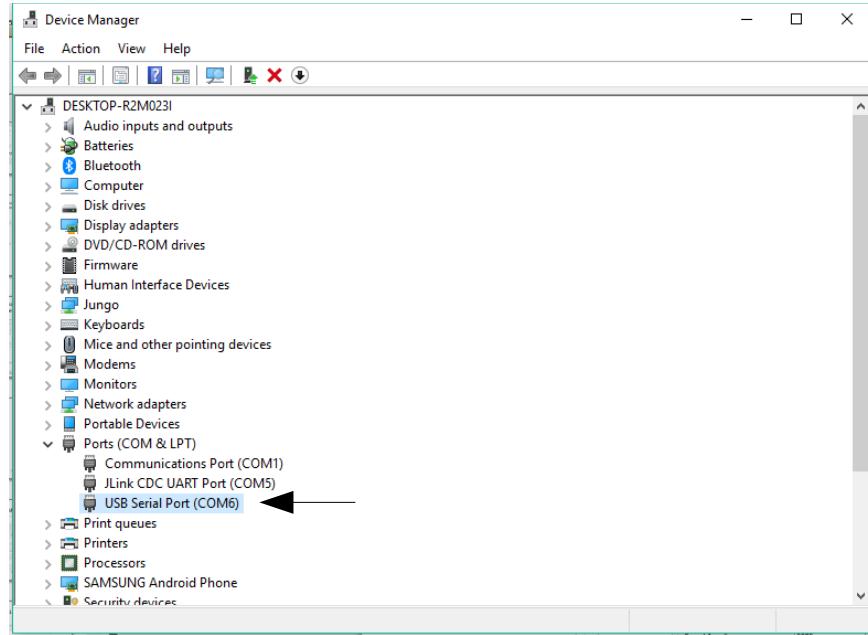
14:39:08 Build Finished. 0 errors, 2 warnings. (took 10s.867ms)

```

gw002\_example\_uart\_rtos\_v1d



# Serial Terminal Program Setting



Test Result

UART by Michael Li