Avnet Visible Things
GPIO Type 1 12 pin Example
by
Michael C. Li
https://www.miketechuniverse.com/
4/4/2018

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 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

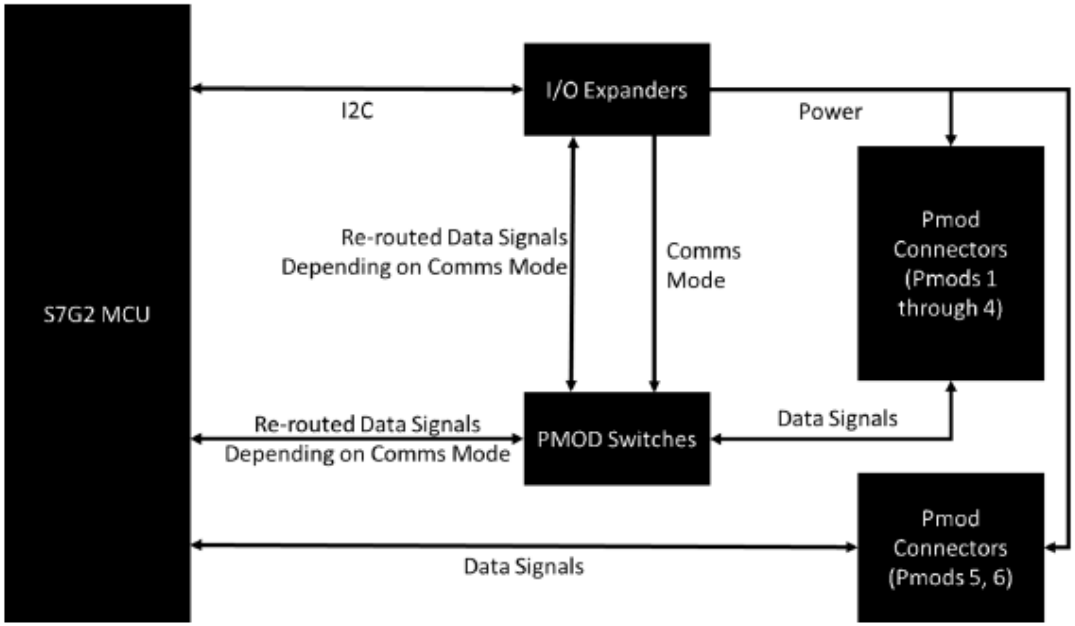
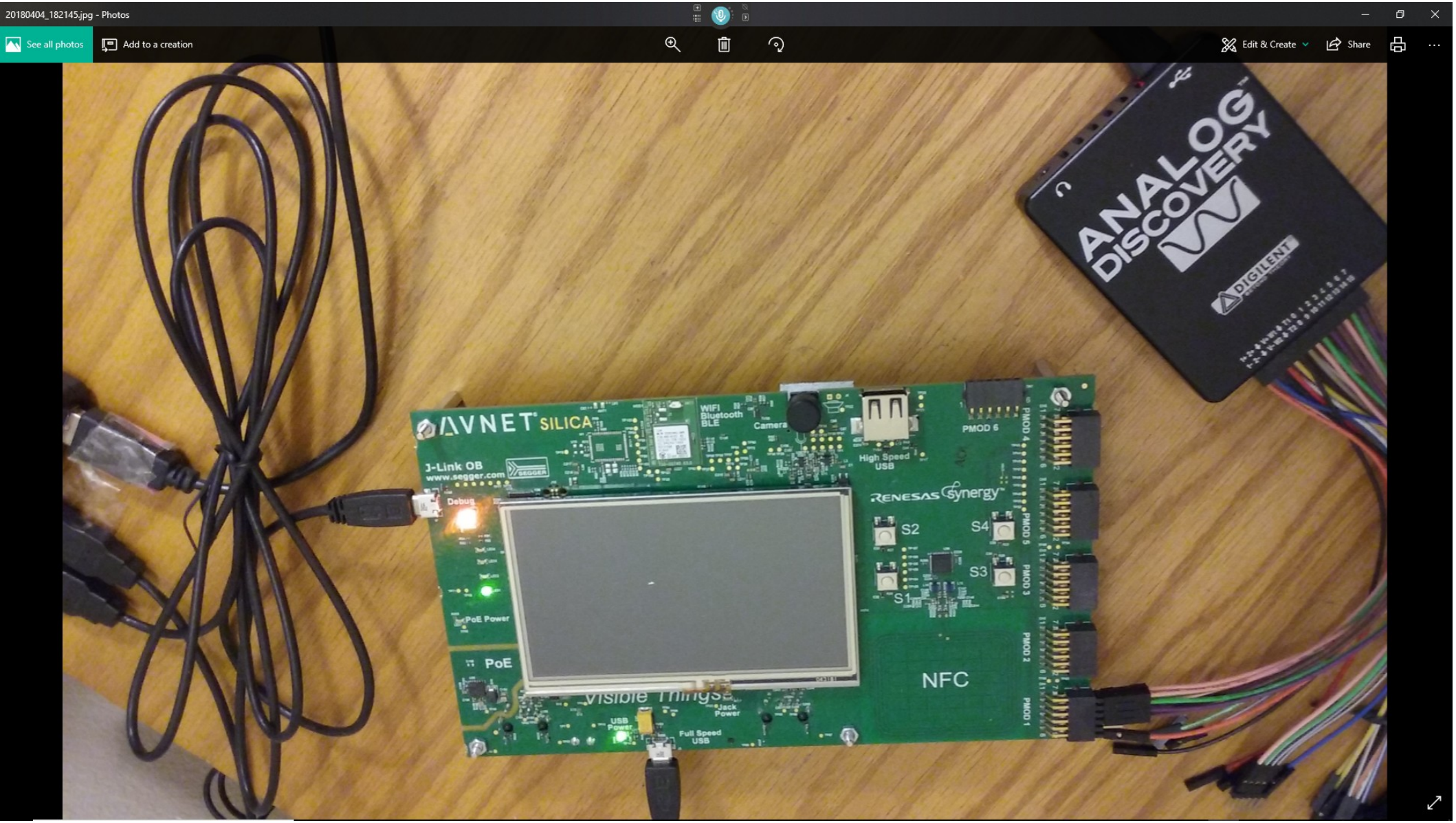
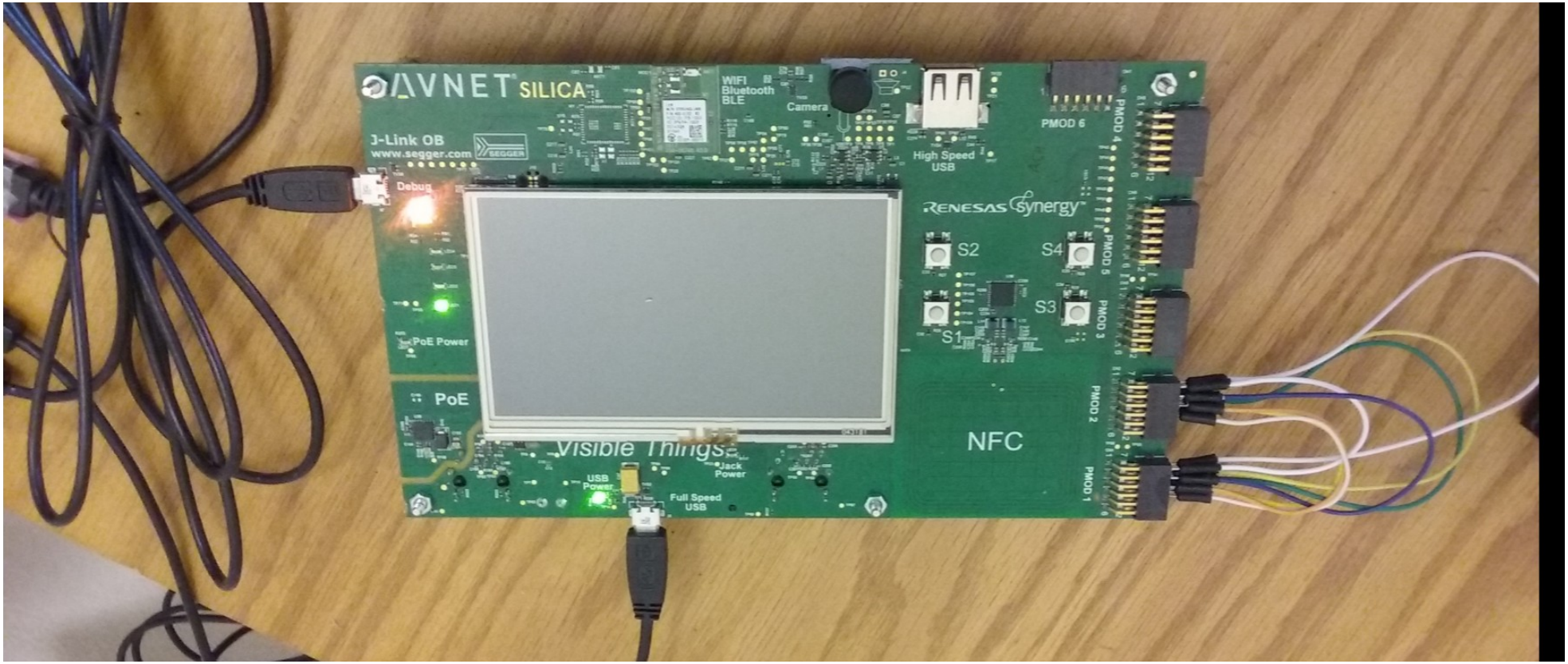


Figure 30 – Pmod, I/O Expander Interconnections Diagram

Test: Probe PMOD ports.



Actual Hardware Setup (Connecting two PMOD ports)



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

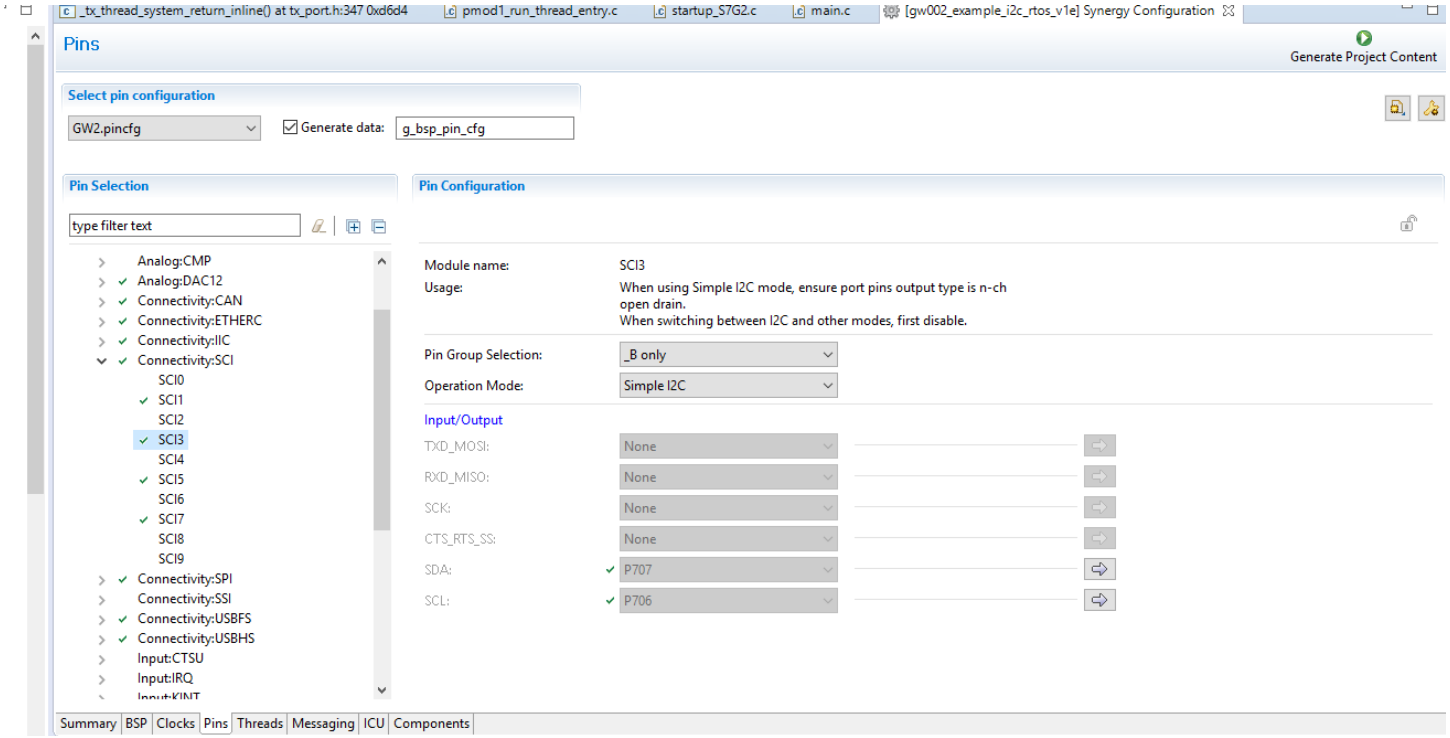
GPIO Type 1 Pmods

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!**

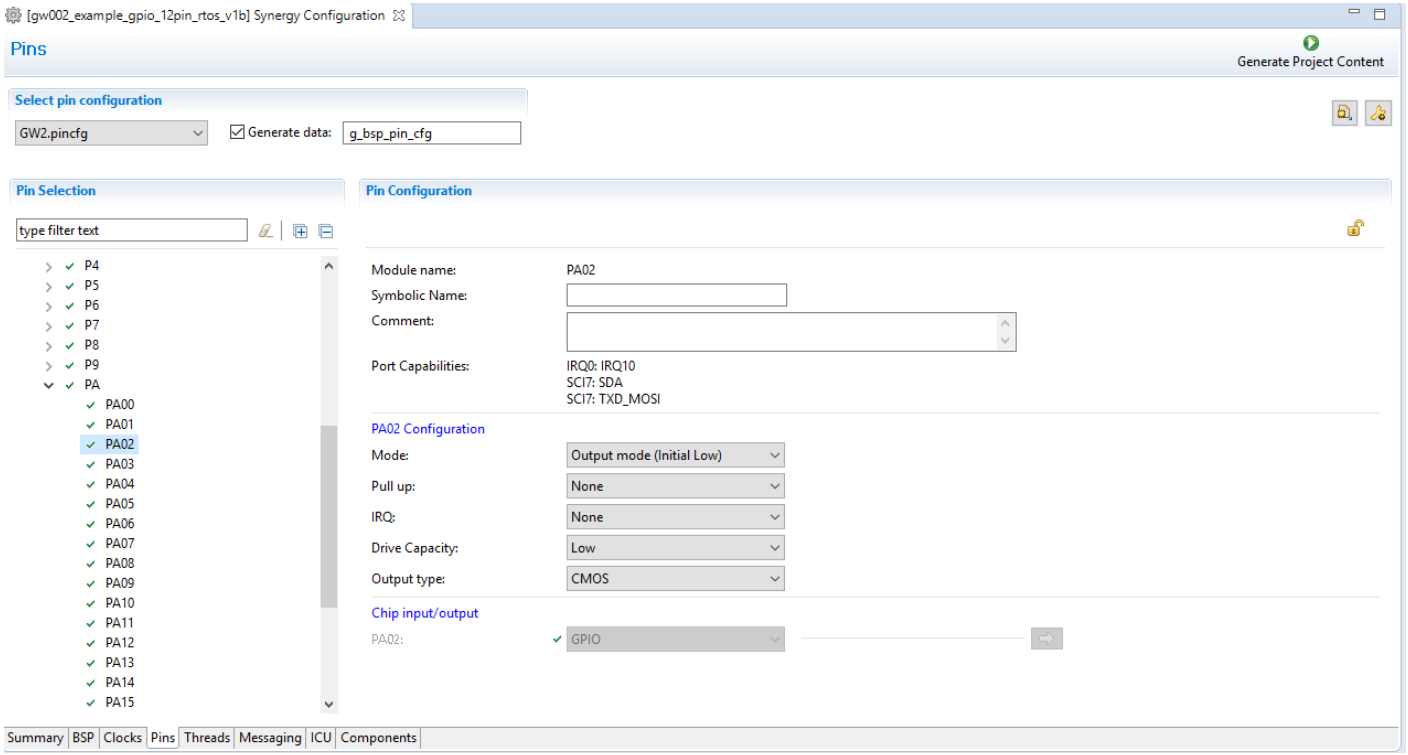
	Pmod		S7G2	
	Pin	Pin Function	Port Pin	Pin Function
Upper Row	1	GPIO	PB02	GPIO
	2	GPIO	PB04	GPIO
	3	GPIO	PB05	GPIO
	4	GPIO	PB03	GPIO
	5	GND	GND	GND
	6	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)
Lower Row	1	GPIO	P001	GPIO
	2	GPIO	I/O Expander	GPIO
	3	GPIO	I/O Expander	GPIO
	4	GPIO	I/O Expander	GPIO
	5	GND	GND	GND
	6	VCC (3.3V)	VCC (3.3V)	VCC (3.3V)

Table 14 – Pmod2 GPIO Type 1 Pin Connections (SPI/UART Mode)

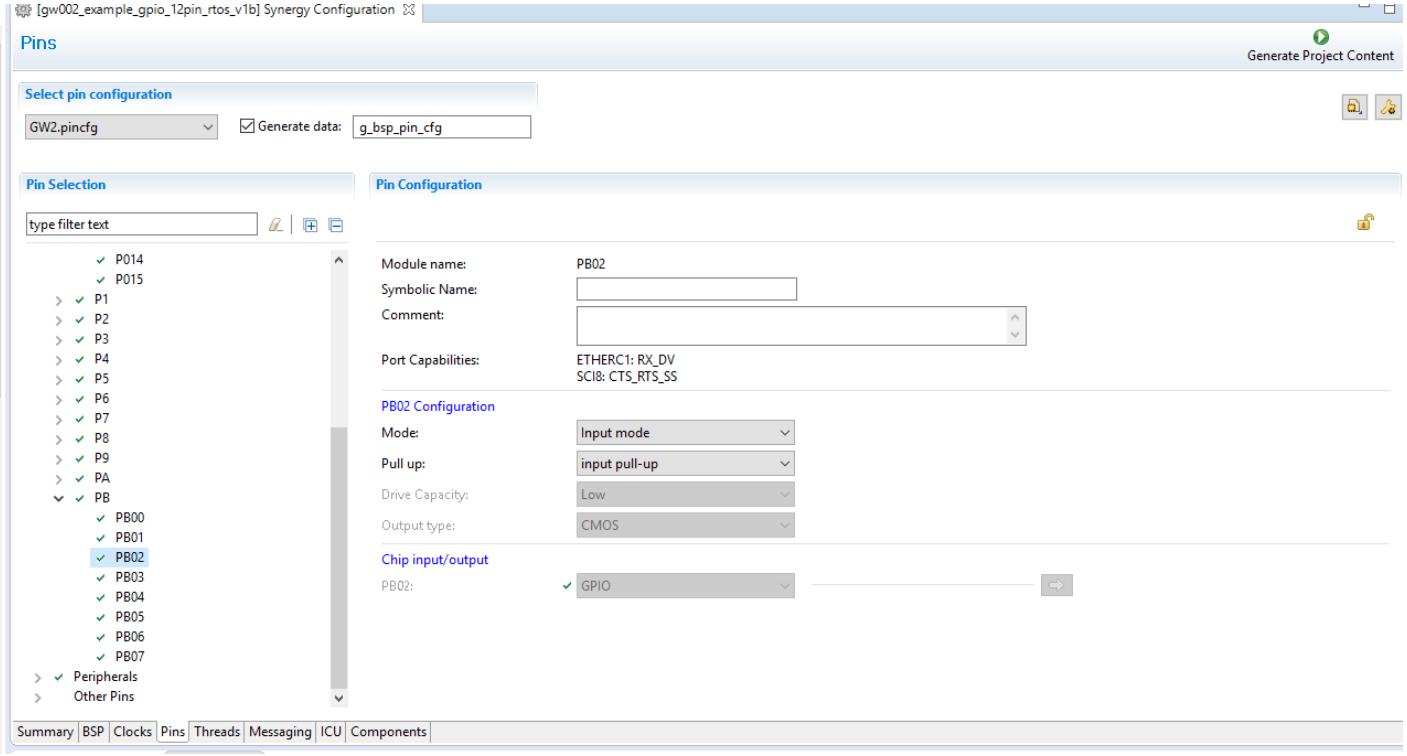
IO Expanders U18/U19 I2C port



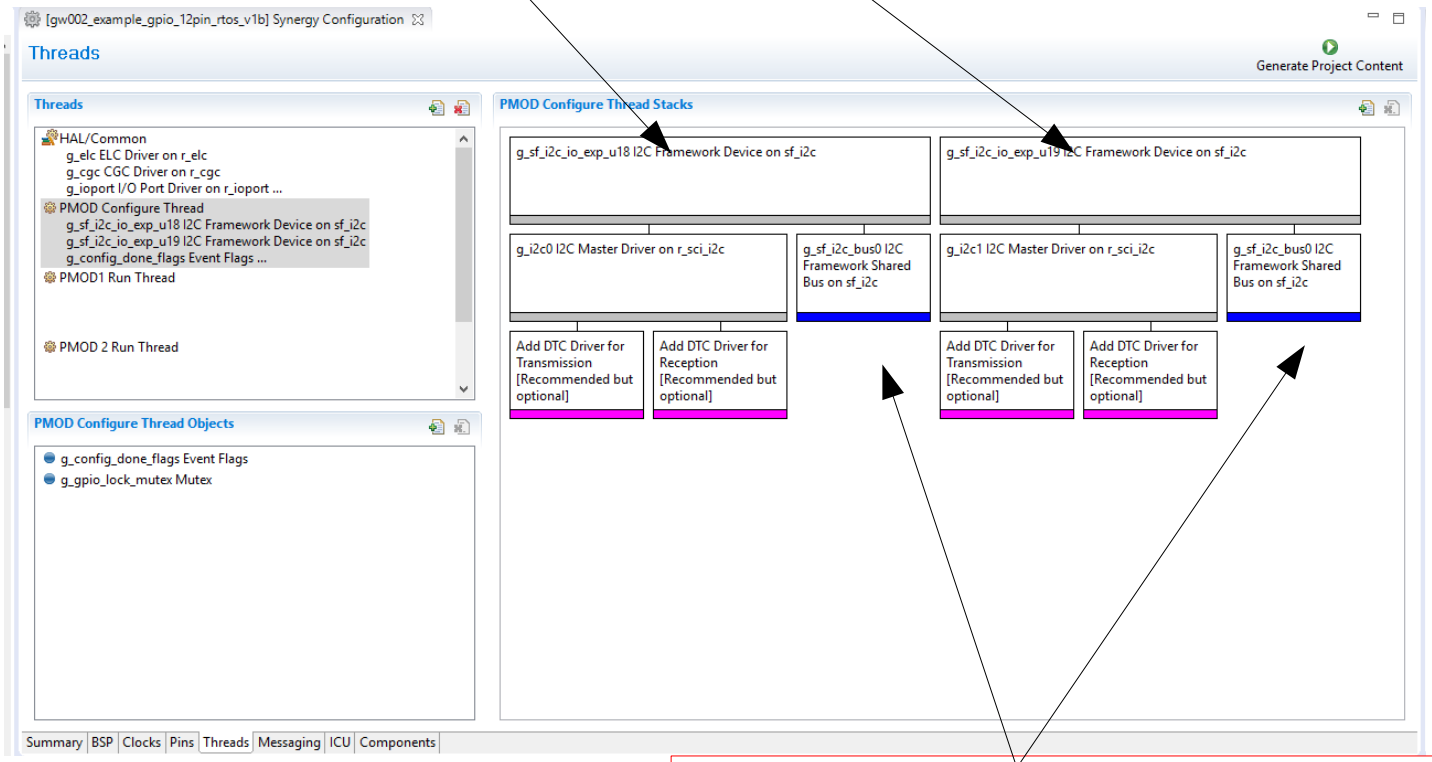
PMOD 1 : PA02/3/4/5 and P400 Output Mode



PMOD 2 : PB02/3/4/5 and P001 Input Mode

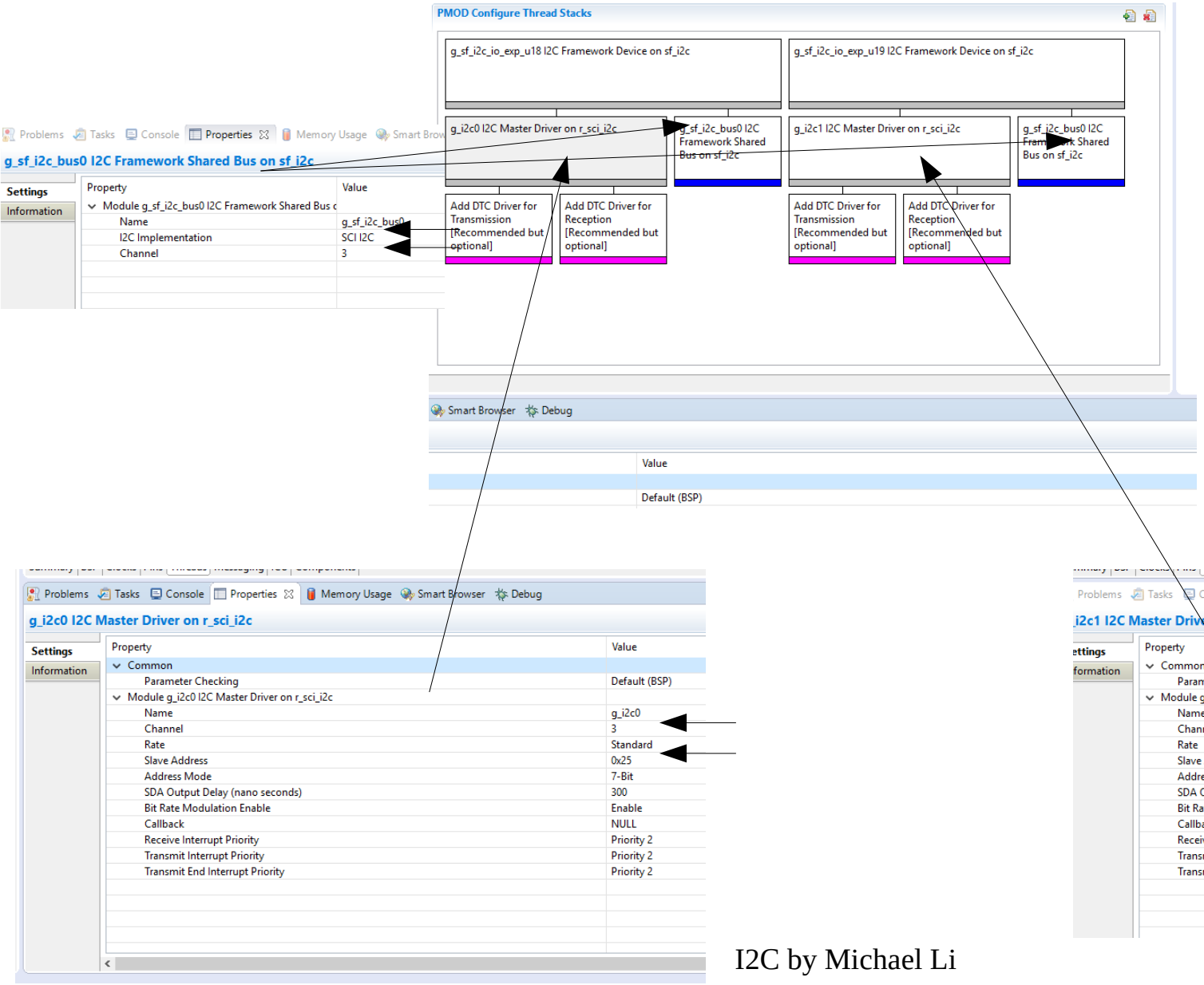


PMOD Configure Thread
(Create U18/U19 I2C Framework Drivers)



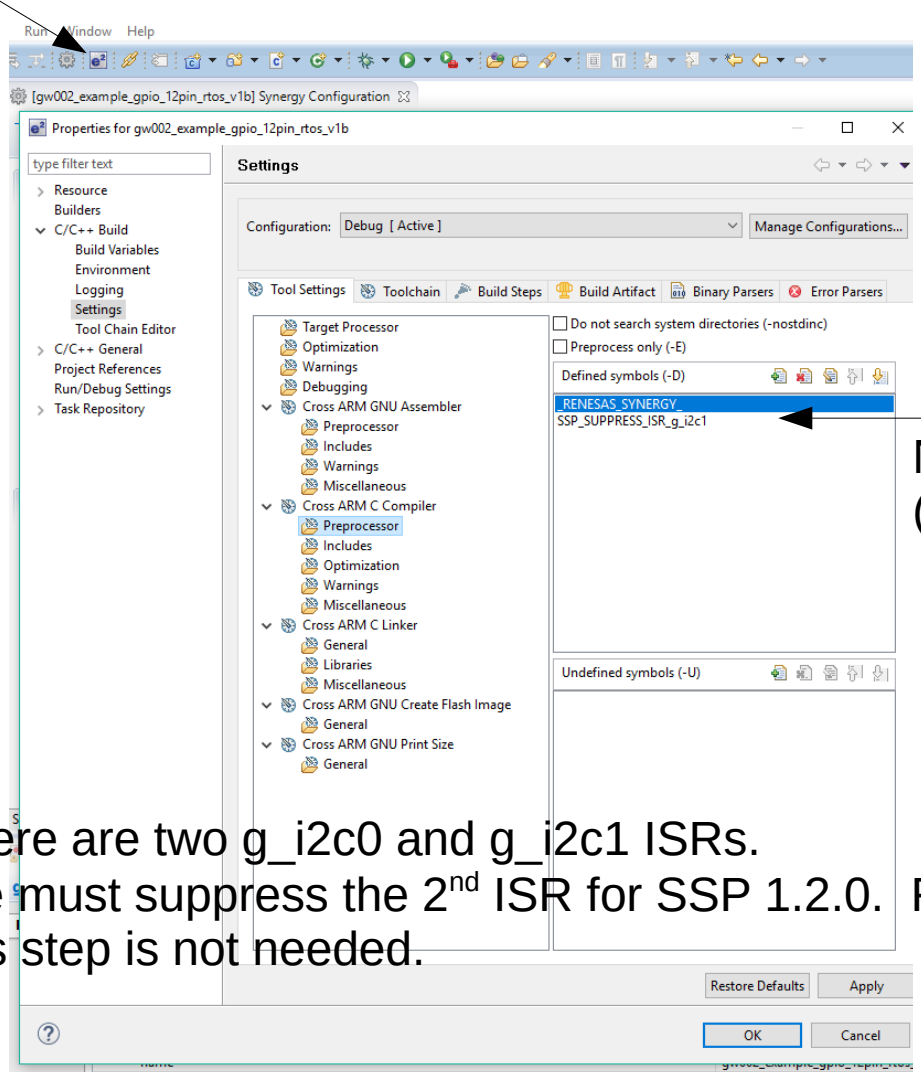
Both drivers share the same g_sf_i2c_bus0 bus.

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
Add DTC Driver for Transmission [Recommended but optional]
Add DTC Driver for Reception [Recommended but optional]
Problems Tasks Console Properties Memory Usage Smart Browser Debug
g_sf_i2c_bus0 I2C Framework Shared Bus on sf_i2c
Settings
Information
Property
Value
Module g_sf_i2c_bus0 I2C Framework Shared Bus
Name
g_sf_i2c_bus0
I2C Implementation
SCI I2C
Channel
3
Default (BSP)
g_i2c0 I2C Master Driver on r_sci_i2c
Common
Parameter Checking
Default (BSP)
Module g_i2c0 I2C Master Driver on r_sci_i2c
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
i2c1 I2C Master Driver on r_sci_i2c
Module g_i2c1 I2C Master Driver on r_sci_i2c
g_i2c1
0x27

Compiler Pre-processor Requirement

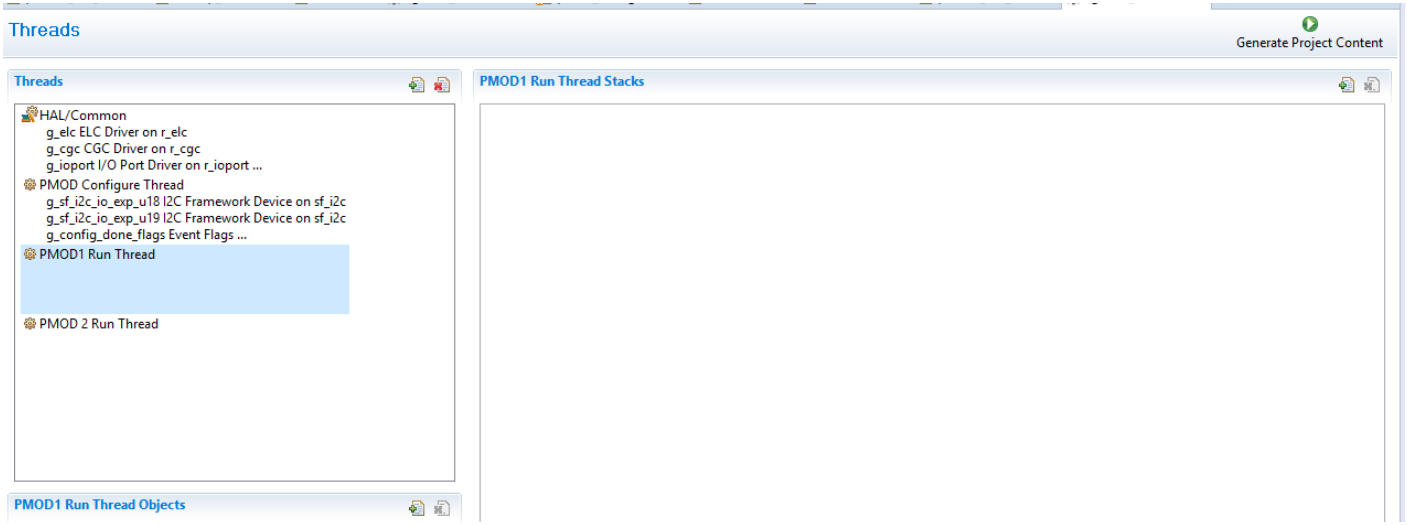
Click this icon!

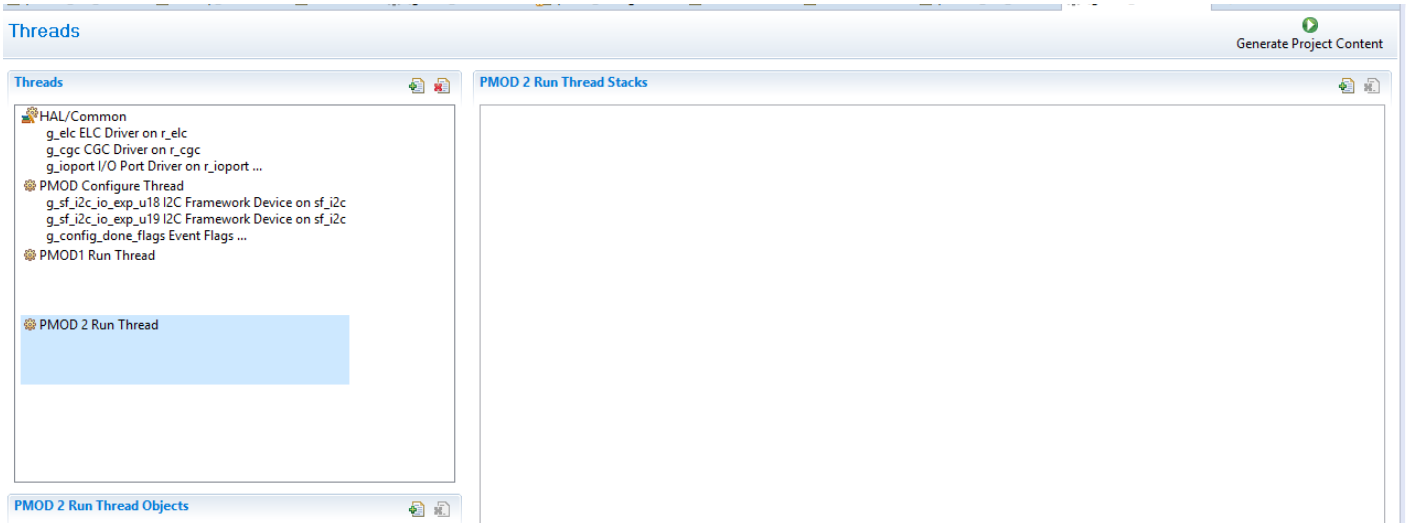


Need to Suppress 2nd i2c ISR (needed for SSP 1.2.0 only)

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.

Both PMOD 1 and 2 Threads have no drivers needed because the ports are configured as GPIO ports.





Configure PMOD 1 and PMOD 2

[gw002_example_gpio_12pin_rtos_v1b] Synergy Configuration | pmod1_run_thread_entry.c | pmod_configure_thread_entry.c

```c
    ////////////////////////////////////////////////
    ///  U18 setup for PMOD1 : GPIO Type 1 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]);   // Low for GPIO Type 1

    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 : GPIO Type 1 12 pin configuration
    ///                        Set IO1-5 output pins (S7G2 MCU pins)
    ///                        Set IO6/7/8 input pins
    ///                        Set IO6/7/8 = 010
    ///
    /// Note IO5 (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]);   // Low for GPIO Type 1

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

    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.
```

PMOD 1 and 2 Threads' entry code

gpio_12pin_rtos_v1b] Synergy Configuration | pmod1_run_thread_entry.c | pmod_configure_thread_entry.c | PCA9535.c | pmod2_run_thread

```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);


     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);
         }
     }

}
```

Writing data to the port.

gpio_12pin_rtos_v1b] Synergy Configuration | pmod1_run_thread_entry.c | pmod_configure_thread_entry.c | PCA9535.c | pmod2_run_thread_entry.c

```c
#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 readdata[8];  // storing the read data

    tx_event_flags_get(&g_config_done_flags, IOEXP_DONE_EVENT_FLAG, TX_AND, &event_flags, TX_WAIT_FOREVER);  // Don't clea

     while (true)
     {
         for (int i=0; i<8; i++) {
             tx_mutex_get(&g_gpio_lock_mutex, TX_WAIT_FOREVER);
             read_pmode_gpio_type1_byte_port (2, readdata+i,pmod_bus_type_cfg[PMOD2_PORT]);
             tx_mutex_put(&g_gpio_lock_mutex);
             tx_thread_sleep(10);
         }
     }

}
```

Reading data from the port.

Since PMOD1/2 ports are connected, the same write data to PMOD 1 port will expect to be received by PMOD 2 thread.

Successful Build

C/C++ - gw002_example_gpio_i2c_spi_uart_v1h/src/pmod1_run_thread_entry.c - e2 studio

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

Project Explorer

- gw002_example_gpio_12pin_rtos_v1b [Debug]
- gw002_example_gpio_6pin_rtos_v1a
- gw002_example_gpio_i2c_spi_uart_v1h
- gw002_example_i2c_rtos_v1e
- gw002_example_spi_rtos_v1a
- gw002_example_uart_rtos_v1d

[gw002_example_gpio_i2c_spi_uart_v1... | startup_S7G2.c | main.c | pmod4_run_thread_entry.c | tx_port.h | pmod1_run_thread_entry.c

```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];

#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_type1_byte_port (1, ~g_pmod2_port_data,pmod_bus_type_cfg[PMOD1_PORT]);
             tx_mutex_put(&g_gpio_lock_mutex);
             tx_thread_sleep(refreshleddelay);

         }
     }

}
```

Outline

- pmod1_run_thread.h
- pmod_configure_thread_entry.h
- pca9535/pca9535.h
- pmod_bus_type_cfg : PMOD_BUS_TYPE_t[]
- refreshleddelay
- g_pmod2_port_data : uint8_t
- pmod1_run_thread_entry(void) : void

Problems | Tasks | Console | Properties | Memory Usage | Smart Browser

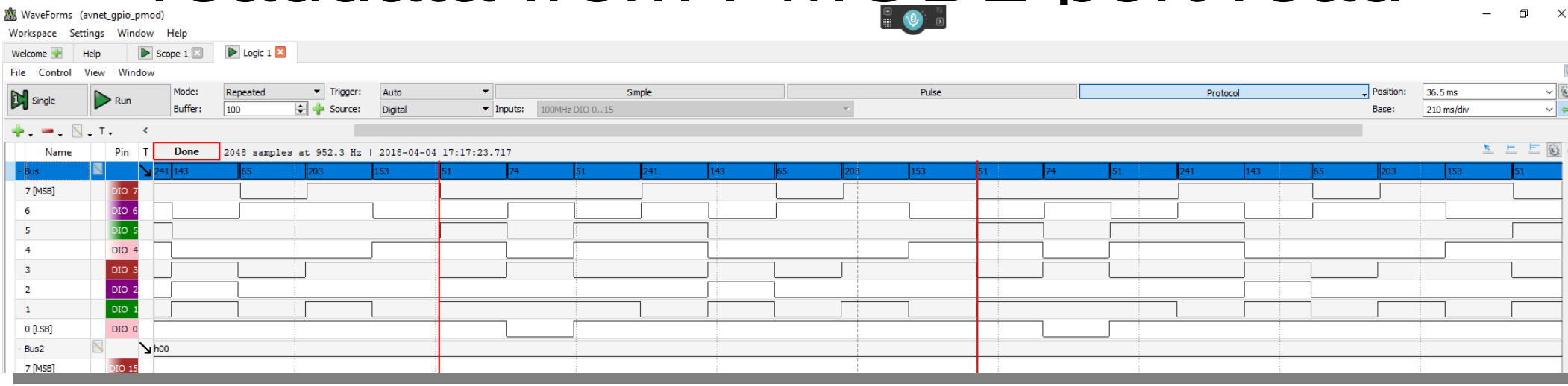
CDT Build Console [gw002_example_gpio_12pin_rtos_v1b]

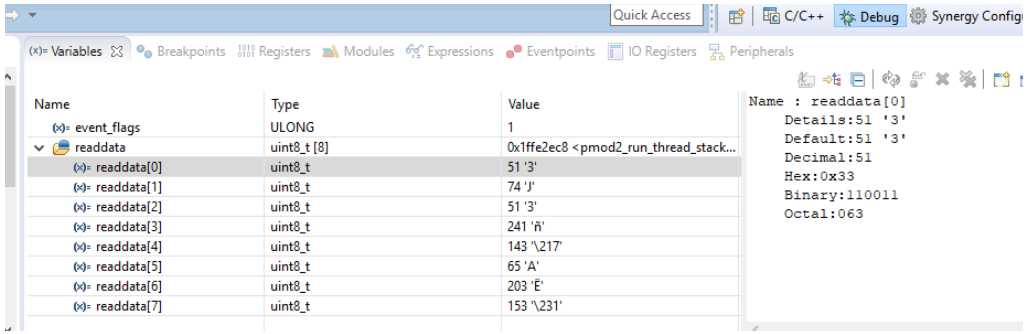
```
'Finished building target: gw002_example_gpio_12pin_rtos_v1b.elf'
' '
'Invoking: Cross ARM GNU Create Flash Image'
'Invoking: Cross ARM GNU Print Size'
arm-none-eabi-objcopy -O srec "gw002_example_gpio_12pin_rtos_v1b.elf"  "gw002_example_gpio_12pin_rtos_v1b.srec"
arm-none-eabi-size --format=berkeley "gw002_example_gpio_12pin_rtos_v1b.elf"
   text    data     bss     dec     hex filename
  49024     268   12812   62104    f298 gw002_example_gpio_12pin_rtos_v1b.elf
'Finished building: gw002_example_gpio_12pin_rtos_v1b.siz'
'Finished building: gw002_example_gpio_12pin_rtos_v1b.srec'
' '
' '

14:09:42 Build Finished. 0 errors, 2 warnings. (took 9s.434ms)
```

gw002_example_gpio_12pin_rtos_v1b

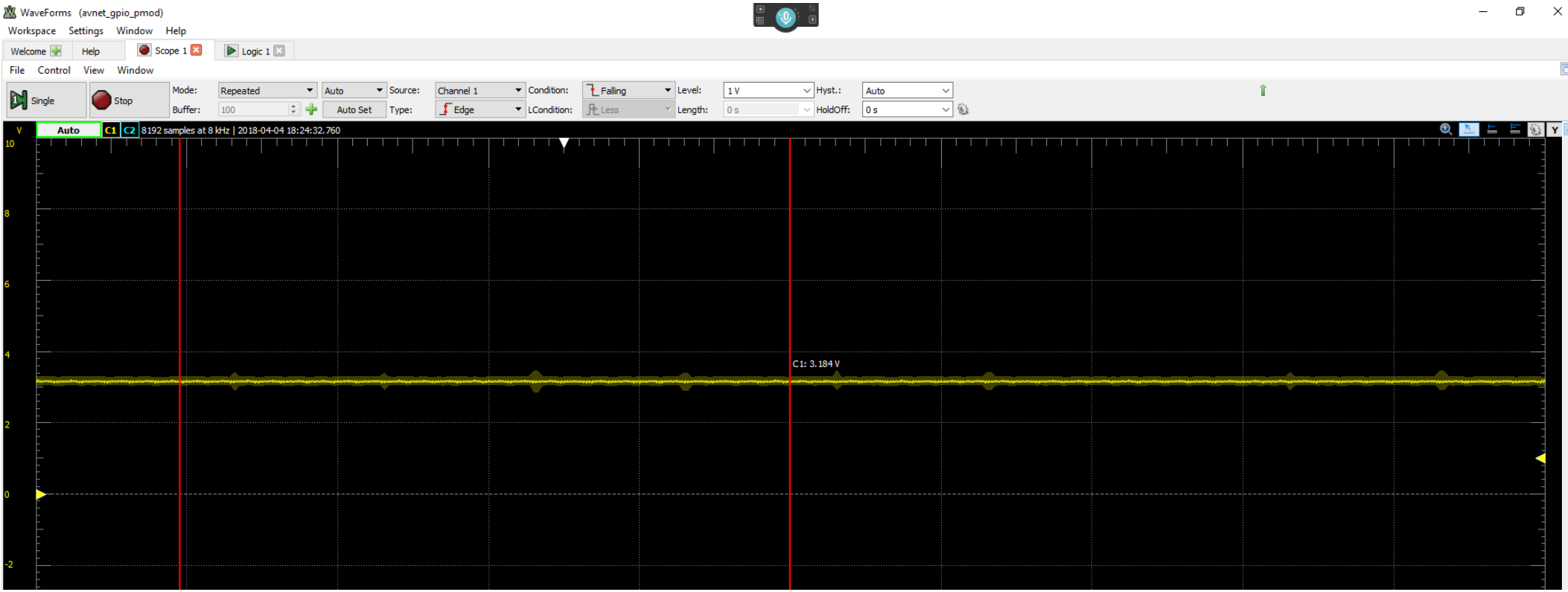
PMOD 1 port scope output data = readdata from PMOD2 port read





Decimal values

PMOD 1 VCC


WaveForms (avnet_gpio_pmod)
Workspace Settings Window Help
Welcome Help Scope 1 Logic 1
File Control View Window
Single Stop
Mode: Repeated Auto Source: Channel 1 Condition: Falling Level: 1 V Hyst.: Auto
Buffer: 100 Auto Set Type: Edge LCondition: Less Length: 0 s HoldOff: 0 s
Auto C1 C2 8192 samples at 8 kHz | 2018-04-04 18:24:32.760
V
10
8
6
4
2
0
-2
C1: 3.184 V