CE211423 - CY8CKIT-042-BLE – F-RAM™ I²C

Objective

This example project is based on a PSoC® Creator™ starter design for the PSoC 4 device. It demonstrates how an F-RAM device can be used with a PSoC device to read and write data. In this project, the PSoC device scans though the F-RAM address space writing the address and then reading it back.

Development Kit Configuration

This example project is designed to run on the CY8CKIT-042-BLE kit from Cypress Semiconductor. A description of the kit, along with more example programs and ordering information, can be found at http://www.cypress.com/documentation/development-kitsboards/cy8ckit-042-ble-bluetooth-low-energy-ble-pioneer-kit.

The project requires configuration settings changes to run on other kits from Cypress Semiconductor. Table 1 lists supported kits. To use any other kit, change the project's device with the help of Device Selector called from the project's context menu.

Table 1. Development Kits and Parts

Development Kit	Device
CY8CKIT-042-BLE	CY8C4247LQI-BL483
CY8CKIT-044	CY8C4247AZI-M485
CY8CKIT-046	CY8C4248BZI-L489

Pin assignments for the supported kits are in Table 2.

Table 2. Pin Assignments for Supported Kits

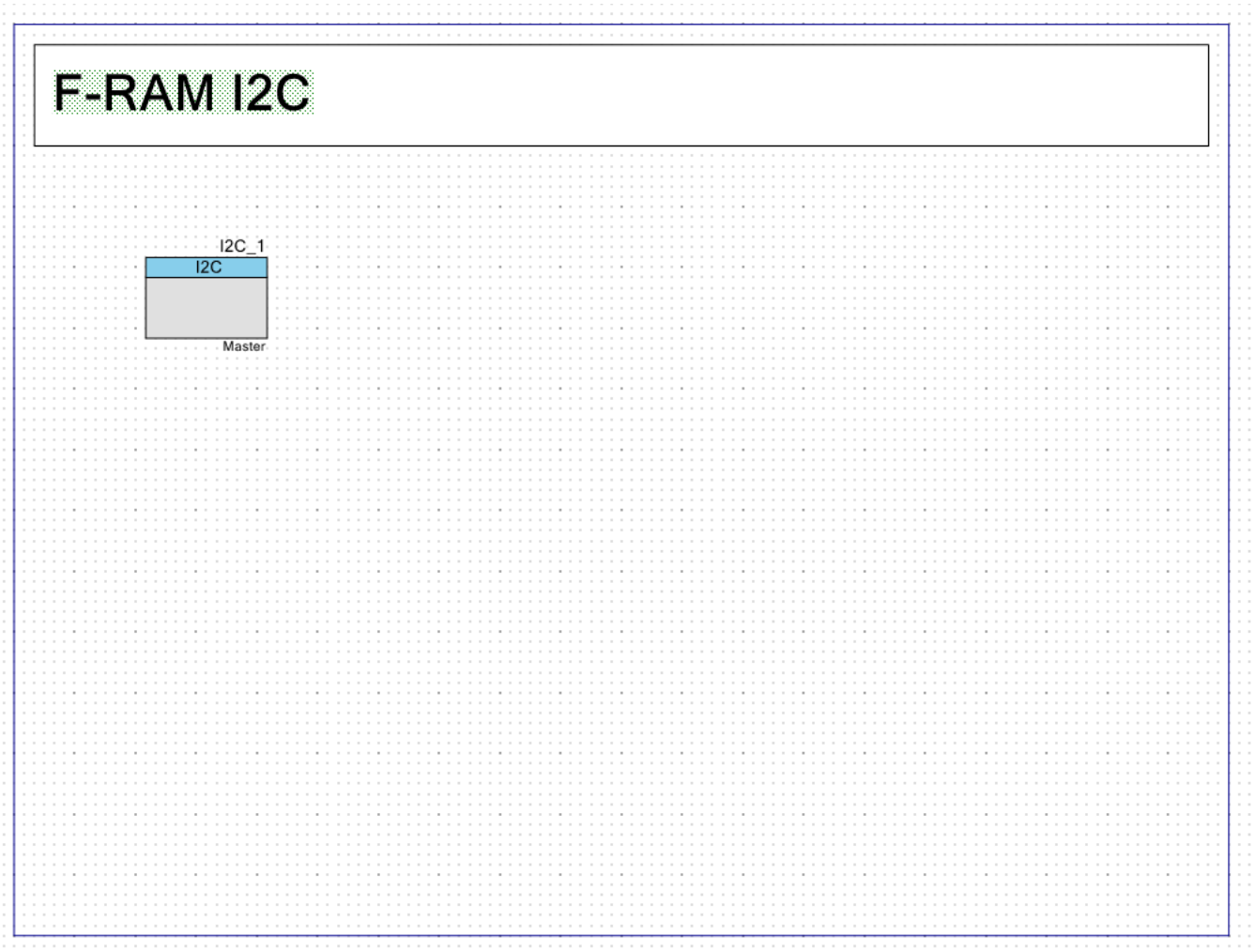
Pin Name	Development Kit			
	CY8CKIT-042	CY8CKIT-042 BLE	CY8CKIT-044	CY8CKIT-046
\I2C_1:scl\	P5[1]	P5[1]	P5[1]	P5[1]
\I2C_1:sda\	P5[0]	P5[0]	P5[0]	P5[0]

Set jumper J9 (J16 for CY8CKIT-042-BLE) to 3.3V position.

Project Configuration

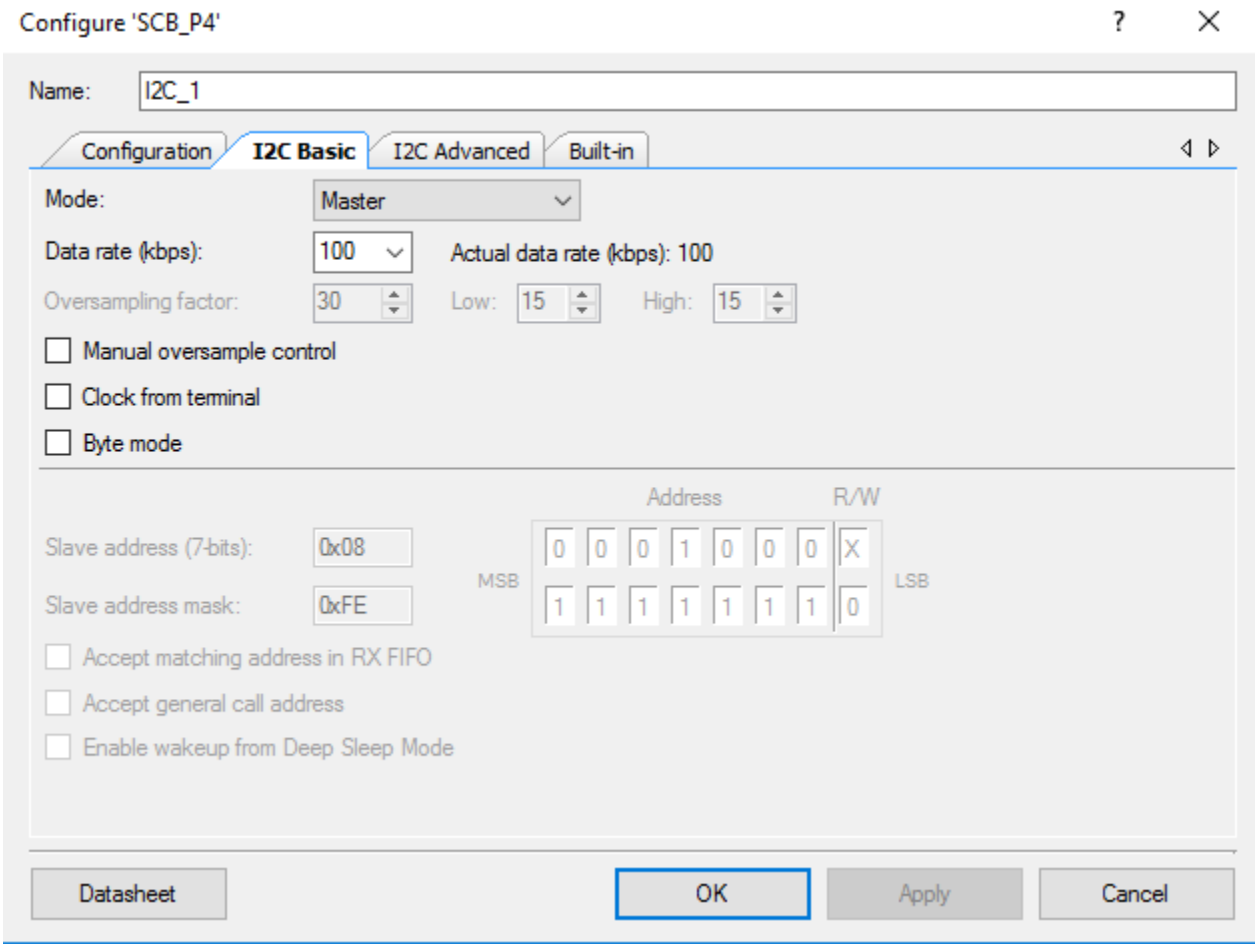
This example project consists of an I²C master and a slave F-RAM device. The top design schematic is shown in Figure 1.

Figure 1. Top Design Schematic



The I²C master is used to move data to and from the F-RAM device. The I²C Component configuration is shown in Figure 2.

Figure 2. I²C Component Configuration Window



Project Description

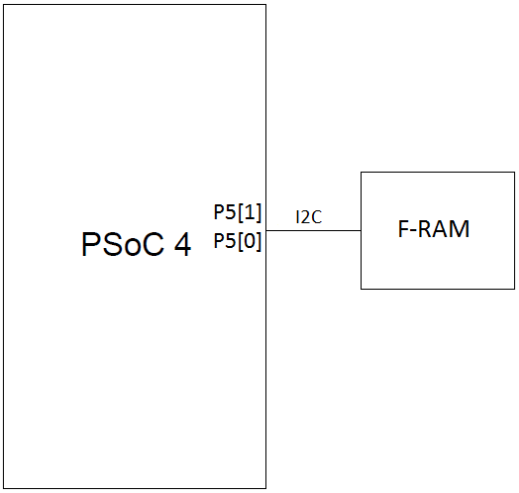
In the main function, the I²C master is started. The “for” loop in *main.c* loops the address space of the F-RAM and writes the address value. . Please refer to the project files included with this project.

Expected Results

There is no output from the PSoC device. This project is designed to be a starting template to show how the F-RAM device is accessed via the PSoC device.

Schematic

Figure 3. Connection Schematic



Using I²C to Communicate with the F-RAM Device

This example project communicates with the F-RAM device using I²C. I²C is a simple Clock/Data bus using 8-bit serial words. Code 1 and Code 2 perform the Write and Read operations for the I²C master to access the F-RAM device.

Code 1. I²C F-RAM Write

```
/*******************************************************************************

* Function Name: FRAM_Write
********************************************************************************
*
* Summary:
*  F-RAM Byte Write Command
* uint16 address - address in the F-RAM array
* uint8 data - data to be written
*
*******************************************************************************/
void FRAM_Write(uint16 address, uint8 data)
{
    uint8 txBuffer[BUFFER_SIZE];

    txBuffer[0] = address >> 8;
    txBuffer[1] = address;
    txBuffer[2] = data;


    /* Clear any previous status */
    I2C_1_I2CMasterClearStatus();

    /* I2C Wrtie Command */
    status = I2C_1_I2CMasterWriteBuf(SLAVE_ADDRESS, (uint8 *) txBuffer, BUFFER_SIZE,
I2C_1_I2C_MODE_COMPLETE_XFER);

    for(;;)
    {
        if(0u != (I2C_1_I2CMasterStatus() & I2C_1_I2C_MSTAT_WR_CMPLT))
        {
            /* Transfer complete. Check Master status to make sure that transfer
            completed without errors. */
            break;
```

```
        }
    }
}
```

Code 2. I²C F-RAM Read

```
/*******************************************************************************
* Function Name: FRAM_Read
********************************************************************************
*
* Summary:
*  F-RAM Byte Read Command
* uint16 address - address in the F-RAM array
* uint8 retrun - data to be read
*
*******************************************************************************/
uint8 FRAM_Read(uint16 address)
{
    uint8 rxBuffer[BUFFER_SIZE-2];
    uint8 txBuffer[2];

    txBuffer[0] = address >> 8;
    txBuffer[1] = address;

    /* Clear any previous status */
    I2C_1_I2CMasterClearStatus();

    /* I2C Wrtie Command to Set the Address*/
    status = I2C_1_I2CMasterWriteBuf(SLAVE_ADDRESS, (uint8 *) txBuffer, 0x2,
I2C_1_I2C_MODE_COMPLETE_XFER);
    for(;;)
    {
        if(0u != (I2C_1_I2CMasterStatus() & I2C_1_I2C_MSTAT_WR_CMPLT))
        {
            /* Transfer complete. Check Master status to make sure that transfer
            completed without errors. */
            break;
        }
    }

    /* Clear any previous status */
    I2C_1_I2CMasterClearStatus();


    /* I2C Read Command */
    status = I2C_1_I2CMasterReadBuf(SLAVE_ADDRESS, (uint8 *) rxBuffer, 1, I2C_1_I2C_MODE_COMPLETE_XFER);
    for(;;)
    {
        if(0u != (I2C_1_I2CMasterStatus() & I2C_1_I2C_MSTAT_RD_CMPLT))
        {
            /* Transfer complete. Check Master status to make sure that transfer
            completed without errors. */
            break;
        }
    }
    return rxBuffer[0];
}
```

Document History

Document Title: CE211423 - CY8CKIT-042-BLE – F-RAM™ I²C

Document Number: 002-11423

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5160314	JLTO	03/23/2016	New code example
*A	5734122	AESATMP9	05/11/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709