SCB_I2cCommMaster Example Project

1.20

Features

- Communication between I^{2}C master and slave
- Simple packet protocol with command and status byte

General Description

This example project demonstrates the basic operation of the I^{2}C master (SCB mode) component. The I^{2}C master sends the packet with a command to the I^{2}C slave to control the RGB LED color. The packet with a status is read back.

Development Kit Configuration

This example project is designed to run on the CY8CKIT-042 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/go/cy8ckit-042.

The second kit is required to implement the I^{2}C slave device to communicate with the master. The SCB_I2cCommSlave example project is provided for this purpose. Refer to the SCB_I2cCommSlave example project datasheet for more information.

The project requires configuration settings changes to run on other kits from Cypress Semiconductor. Table 1 is the list of the supported kits. To switch from CY8CKIT-042 to 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 vs Parts

Development Kit	Device
CY8CKIT-040	CY8C4014LQI-422
CY8CKIT-041	CY8C4045AZI-S413 / CY8C4146AZI-S433
CY8CKIT-042	CY8C4245AXI-483
CY8CKIT-042-BLE	CY8C4247LQI-BL483
CY8CKIT-044	CY8C4247AZI-M485
CY8CKIT-046	CY8C4248BZI-L489
CY8CKIT-048	CY8C4445AZI-483

The pin assignments for the supported kits are in Table 2.

Table 2. Pin Assignment

Development Kit	Pin Name			
	\I2CM:scl\	\I2CM:sda\	LED_RED	LED_GREEN
CY8CKIT-040	P1[2]	P1[3]	P3[2]	P1[1]
CY8CKIT-041	P3[0]	P3[1]	P3[4]	P2[6]
CY8CKIT-042	P3[0]	P3[1]	P1[6]	P0[2]
CY8CKIT-042 BLE	P3[5]	P3[4]	P2[6]	P3[6]
CY8CKIT-044	P4[0]	P4[1]	P0[6]	P2[6]
CY8CKIT-046	P4[0]	P4[1]	P5[2]	P5[3]
CY8CKIT-048	P4[0]	P4[1]	P1[4]	P2[6]

Note The project control files handle the pins placement automatically according to a selected PSoC.

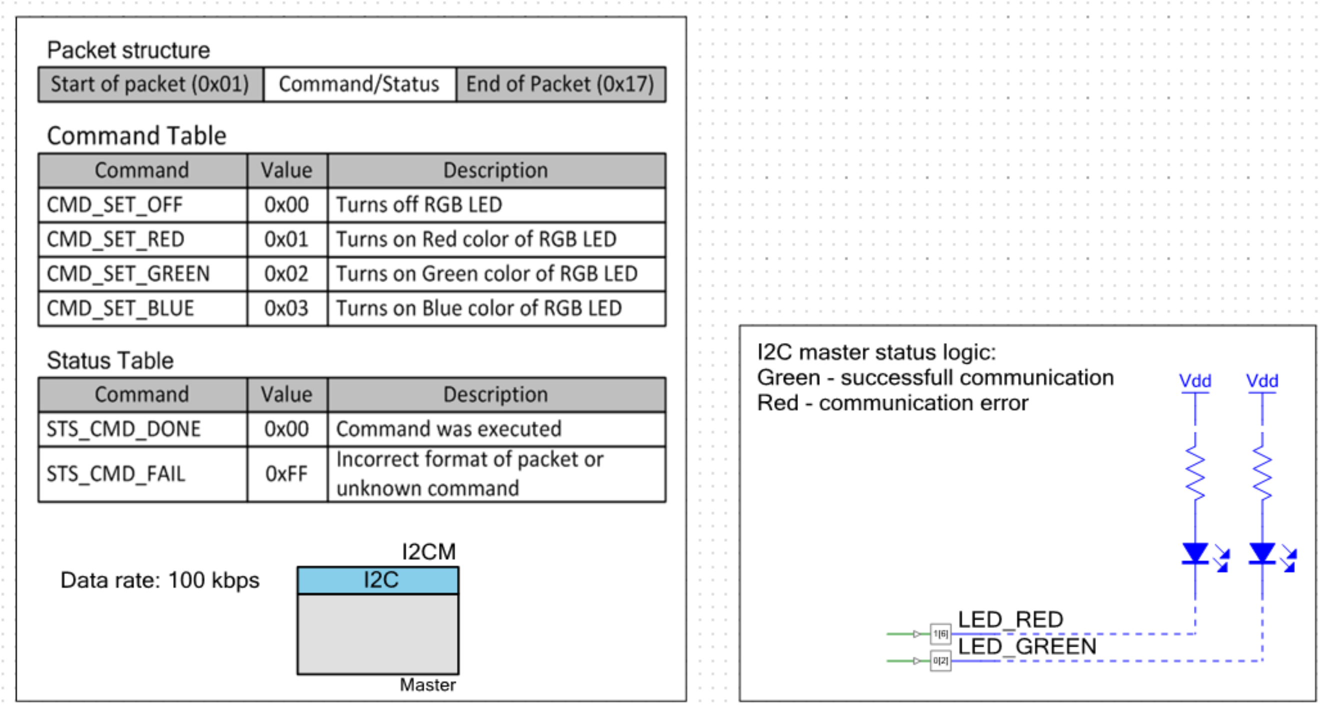
The SCL and SDA lines of the master and slave have to be tied together. These pins and RGB LED pin location for supported kits are summarized in Table 2.

Change the clock configuration, **required only for CY8CKIT-040**. In the Workspace Explorer window, double-click the project's design-wide resource file and click on the **Edit Clocks**… icons on the **Clocks tab**. Set IMO frequency to **32 MHz**.

Project Configuration

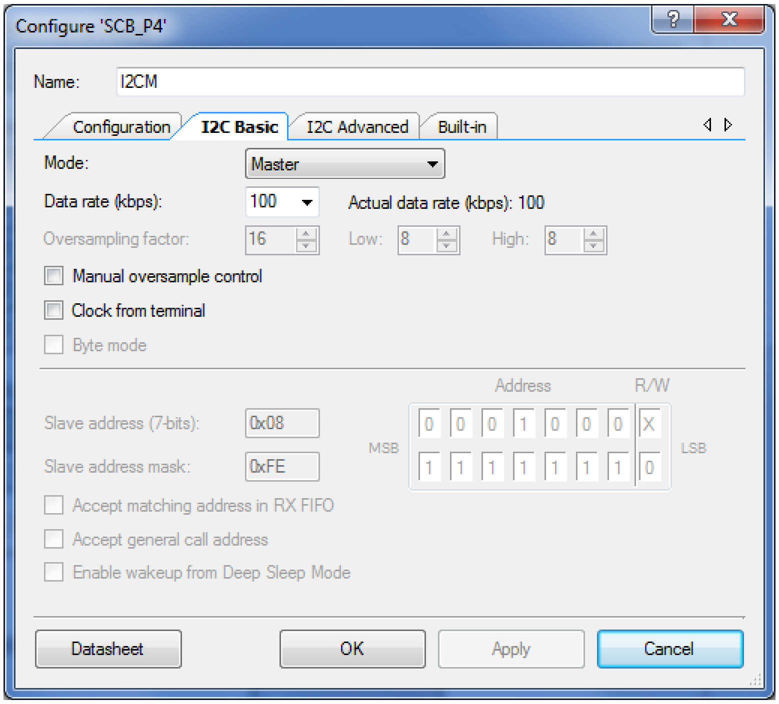
The example project consists of the I²C master (SCB mode) and pin components. The design schematic is shown in Figure 1. The blue annotation components are used to represent the RGB LED installed on the Pioneer Kit. Two pin components are used to control the LED color. The I²C master sends a packet with a command to the slave and reads the back packet with a status every 500 milliseconds.

Figure 1. Example Project Design Schematic



The I^{2}C master is configured to operate with the data rate of 100 kbps. The component configuration window is shown below.

Figure 2. I2C Master (SCB mode) Component Configuration



Project Description

In the main firmware routine, the I^{2}C master component is started. The LED is turned off. Interrupts are enabled to the CPU core as required by the I^{2}C master component operation.

The main loop starts communication with the slave every 500 milliseconds. The I^{2}C master initializes a packet with a command and setups write transfer. The initial command is CMD_SET_RED. The packet structure and table with commands are shown below. The code polls I2CM_I2CMasterStatus() API for a completion of the write transfer. After the write transfer is completed the errors status bit is checked to update the LED color: green – a successful transfer or red – any error occurred during the transfer. In the case of an error the following read transfer does not take place and the same command will be sent again. Otherwise the I^{2}C master allocates a read buffer and setups the read transfer to receive the status packet. The code polls I2CM_I2CMasterStatus() to complete a read transfer. After the read transfer is completed, the errors status bit is checked to update the LED color: green – a successful transfer and red – any error occurred during the transfer. If an error occurs, the same command will be sent again. Otherwise the basic checks of the packet structure are done: the length of the packet, the start and end of the packet byte. The status byte is checked afterwards. If all the checks are successful, the command is considered as executed. The command is updated to the next one. The LED is turned off after write and read transfers have been completed. The command sequence is the following: CMD_SET_RED, CMD_SET_GREEN, CMD_SET_BLUE, CMD_SET_OFF, CMD_SET_RED and so on.

The packet structure

Start of packet (0x01)	Command/Status	End of Packet (0x17)

Table 3. Command constants

Command	Value	Description
CMD_SET_OFF	0	Turns off RGB LED
CMD_SET_RED	1	Turns on Red color of RGB LED
CMD_SET_GREEN	2	Turns on Green color of RGB LED
CMD_SET_BLUE	3	Turns on Blue color of RGB LED

Table 4. Status constants

Status	Value	Description
STS_CMD_DONE	0x00	Command was executed
STS_CMD_FAIL	0xFF	Incorrect format of packet or unknown command

The packets with a command and status are converted into the following I²C master transfers. The packet with a command has a write direction set in the address byte and the packet with a status has a read direction set appropriately.

Packet with command

S	ADDR = 0x08	W	A	SOP = 0x01	A	Command	A	EOP = 0x17	A	P

Packet with status

S	ADDR = 0x08	R	A	SOP = 0x01	A	Status	A	EOP = 0x17	$\overline{A}$	P

- Master drives the bus - Slave drives the bus

Expected Results

Connect two boards as explained in the Development Kit Configuration section.

Build SCB_I2cCommSlave example project and program into the device.

The CY8CKIT-042 kit does not provide a pull-up on the I²C bus unless the Bridge Control panel enables it. If both master and slave examples run on this kit a pull-up has to be enabled. Run the Bridge Control Panel software shipped with the PSoC Creator. Select the KitProg device, which was programmed with SCB_I2cCommSlave example project, from the list of the Connected Ports. Refer to Figure 3.

All other kits except CY8CKIT-042 provide a pull-up on the I²C bus when USB mini-B cable is plugged-in therefore no extra actions are needed before programming the SCB_I2cCommMaster example project.

Build SCB_I2cCommMaster example project and program into the device.

Observe that RGB LED changes its color on the kit which serves as an I²C slave.

The RGB LED on kit which serves as I²C master indicates that command and status sequence is completed successfully by toggling green color. If an error has occurred while transfer the LED turns to red.

Figure 3. Bridge Control Panel Enables Pull-ups on I²C Bus